Microsoft® Exchange 2000 Server Directory Access & Integration with Microsoft® Windows® 2000

PT102
Version 1.1

From the Exchange 2000 Server Pre-emPt Series

Paul Bowden
Program Manager - Europe
Microsoft Exchange Server Product Unit

Pt
PLATINUM

Microsoft Technical Readiness
Service Partner Support Sales

© 1999 Microsoft Corporation. All rights reserved.

The information contained in this White Paper represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

The example companies, organizations, products, people and events depicted herein are fictitious. No association with any real company, organization, product, person or event is intended or should be inferred.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

BackOffice logo, Microsoft, Win32, Windows, and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Other product or company names mentioned herein may be the trademarks of their respective owners.

Microsoft Corporation • One Microsoft Way • Redmond, WA 98052-6399 • USA

Contents

MICROSOFT® EXCHANGE 2000 SERVER DIRECTORY ACCESS & INTEGRATION WITH MICROSOFT® WINDOWS® 2000.....1

CONTENTS.....II

ACKNOWLEDGMENTS.....II

DOCUMENT VISION AND SCOPE.....2

INTRODUCTION.....2

Exchange 2000 Server Integration with Windows 2000.....2
Management of Exchange 2000.....2

ACTIVE DIRECTORY TECHNOLOGY.....2

Domain.....2
Domain Tree.....2
Forest.....2
The First Domain.....2
Domain Controller.....2
Domain Mode.....2
Site.....2
Naming Contexts.....2
Location Service Providers.....2
Global Catalog Servers.....2
Client Log On and Authentication.....2
Per-Property Replication.....2
GUIDs and Distinguished Names.....2

EXCHANGE 2000 SERVER AND ACTIVE DIRECTORY.....2

Introduction.....2
Directory Objects.....2
Consolidating User Management Tools.....2
Planning Active Directory with Exchange 2000 In Mind.....2

THE GLOBAL ADDRESS LIST.....2

Global Address List.....2
Active Directory Domain Names.....2
Determining the Number of Global Catalog Servers Required.....2
Deploying Active Directory Forests.....2
Synchronizing Directory Data Between Forests.....2
Directory Information.....2
Public Folders.....2
Calendaring and Scheduling.....2

RICH DIRECTORY ACCESS.....2

Understanding Selectable Field Replication.....2
Selecting Attributes to Replicate to the Global Catalog.....2
Exchange 2000 Installation and Active Directory Schema.....2
Making Schema Changes.....2

IMPLEMENTING GROUPS IN ACTIVE DIRECTORY.....2

Exchange Server 5.5 Distribution Lists.....2
Active Directory Groups.....2

Recommendations for Implementing Groups....2
Group Creation Decision Process....2
Exchange 2000 Beta 3 Scenarios....2
Coexisting with and Upgrading from Previous Versions of Exchange....2

ADDRESS BOOK LOOKUPS....2

DSProxy....2
Client Access to Active Directory....2
Exchange Client and Outlook 97/98....2
Traffic and Load Generated Through the DSProxy Process....2
Outlook 2000....2
Exchange 2000 Directory Access Cache....2
Manually Adjusting the Cache Parameters....2
Getting Directory Data....2
Message Routing and Group Expansion....2

ADDRESS LISTS AND OFFLINE ADDRESS BOOKS....2

Address Book Views....2
Address Lists....2
Address List Compatibility with Exchange Server 5.*x*....2
Address List Service....2
Offline Address Books....2

APPENDIX A: RFC2254 – LDAP SEARCH FILTERS....2

Appendix B: LDAP Attribute Names in Exchange 5.5 and Exchange 2000....2

Acknowledgments

Many thanks to the following people for their time and enthusiasm. No white paper is produced without many people going out of their way to assist.

Exchange Product Group

Peter Waxman

Neil Koorland

John Freeman

Bogdan-George Pintea

Scott Roberts

Geeman Yip

John Kenerson

Neil Shipp

David Lemson

David Madison

Windows 2000 Product Group

Steven Judd

Don Hacherl

Behrooz Chitsaz

Product Support Services (PSS)

Andy Moss

Microsoft Consulting Services, UK

Peter Nilsson

Julius Davies (Windows 2000 Lead)

Ingrid Robson

Darren Strange

Others

Laura Payne (Exchange Program Manager, Service Readiness)

Andreas Luther (Windows 2000 Program Manager, Deployment Assets)

David Trulli (Principal Consultant, MCS New Jersey)

Everyone on PlatChat

Philip Beville, Richard Minty, and the rest of the Exchange Global Engineering team at Credit Suisse First Boston

Document Vision and Scope

The purpose of this document is to provide you with a better understanding of how Microsoft® Exchange 2000 Server uses Microsoft® Windows® 2000 and Active Directory.

After reading this white paper, you should understand the load placed on the different classes of Active Directory servers. This will enable you to successfully deploy Exchange 2000 by appropriately planning for the resources you will need for Windows 2000. Additionally, this white paper will help you plan and test Active Directory and understand the impact that Exchange 2000 has on scalability models.

This information should be part of your learning cycle for Exchange 2000 Server. To gain a full understanding of the other technologies in the product, you may want to read the other documents in this series:

- PT100 – Windows 2000 and Exchange 2000 Server Terminology Primer
- PT101 – Deploying the Active Directory Connector
- PT103 – Exchange 2000 Server Coexistence and Upgrades
- PT104 – Understanding Message Routing in Exchange 2000 Server
- PT105 – Understanding Exchange 2000 Server Storage Technology
- PT106 – Deploying Exchange 2000 Server Real-time Collaboration Services
- PT107 – Collaboration with Exchange 2000 Server

Note The information in this document is based on Microsoft® Windows® 2000 Release Candidate 2 and Microsoft® Exchange Server 2000 Beta 3.

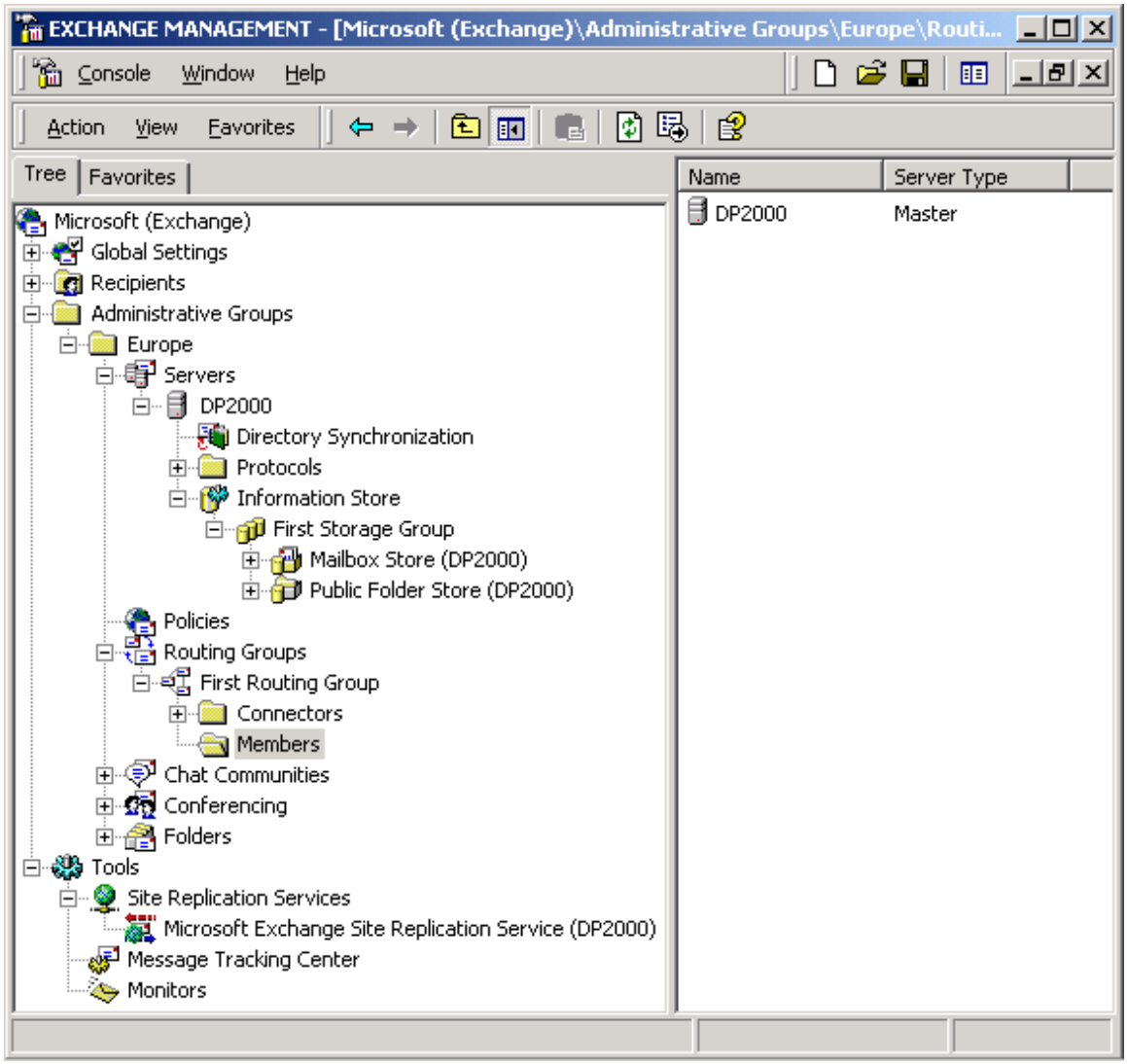
Introduction

Exchange 2000 Server Integration with Windows 2000

Microsoft Exchange 2000 Server relies heavily on Microsoft Windows 2000 in three main areas; the directory, transport, and name resolution.

Directory

Unlike previous versions of Exchange, Exchange 2000 does not include its own directory service. Instead, Active Directory in Windows 2000 provides functionality such as the Global Address List, Address Book Views, and Offline Address Books.



Transport

After you install Windows 2000, base Internet protocol stacks, such as Simple Mail Transfer Protocol (SMTP) and Network News Transfer Protocol (NNTP), are configured as part of the operating system. The operating system and other components use these stacks; for example, it is possible to replicate Active Directory information using SMTP rather than Remote Procedure Calls (RPC). The Exchange 2000 installation extends these

stacks with additional command verbs and advanced routing components to provide all of the functionality required for an enterprise-class messaging and collaboration system. These extensions are installed as a set of *sinks* that are registered with each protocol.

Name Resolution

Exchange 2000 is a network-oriented product that relies heavily on its underlying network infrastructure and protocols to carry data. Clients communicate with servers to access collaboration data, and servers communicate with other servers to route messages. All products of this nature require a solid name resolution method. In Microsoft® Windows® NT 4.0, the Windows Internet Name Service (WINS) provides network basic input/output system (NetBIOS) name resolution and Domain Name System (DNS) for winsock-based name resolution. All previous versions of the Outlook and Exchange clients use, by default, the winsock layer for communications and consequently DNS as the preferred name resolution technique. For Windows 2000, other components such as logon and domain validation have moved over to DNS as the primary name resolution method.

Management of Exchange 2000

The paradigm for administrating Exchange 2000 is slightly different, and is greatly improved from the Exchange Server 5.5 management model used in the past. The term Exchange *site* is no longer used because this concept now belongs to the Active Directory. Instead, the terms *administrative groups* and *routing groups* are used.

An administrative group is a collection of Exchange servers that can be managed as one logical unit. Support staff can manage all of the servers in an administrative group no matter where those servers are located in the organization. The configuration data for the group is held in the configuration naming context of Active Directory, which is fully replicated around the organization and is available for read/write access by any server. Subsequently, a single administrative group is not tied to the characteristics of the underlying network or domain structure. This means that administrators do not need to worry about RPC connectivity or slow links when designing their administrative groups. They can define the groups based on individual business requirements, without any other constraints.

A routing group is a collection of Exchange 2000 servers that send messages to each other directly, without going through a bridgehead. A routing group belongs to an administrative group but it is possible for a server to belong to a routing group that is outside of the server's administrative group.

Note The relationship between administrative groups and routing groups is a little more tied when previous versions of Exchange server are present in the Exchange organization. For more information, see the "PT103 Coexistence and Upgrades" white paper for more information.

Through these new group types, Exchange 2000 becomes easier to administer and more flexible.

1.

Active Directory Technology

Although a full explanation of Active Directory design is outside the scope of this paper, a brief explanation of the terminology and structure of Active Directory is described below. Once you have a basic understanding of the technology, you can examine what Exchange 2000 Server requires out of the Active Directory to ensure an efficient, scalable deployment.

Domain

An Active Directory domain is a unit of scope and administration for a collection of resources on the network. The domain boundary defines the namespace and includes one or more domain controllers. All Active Directory domains are identified by a Domain Name System (DNS) style naming convention as well as a name that is compatible with network basic input/output system (NetBIOS); for example:

DNS-style domain: EUROPE.MICROSOFT.COM

NetBIOS name: EUROPE

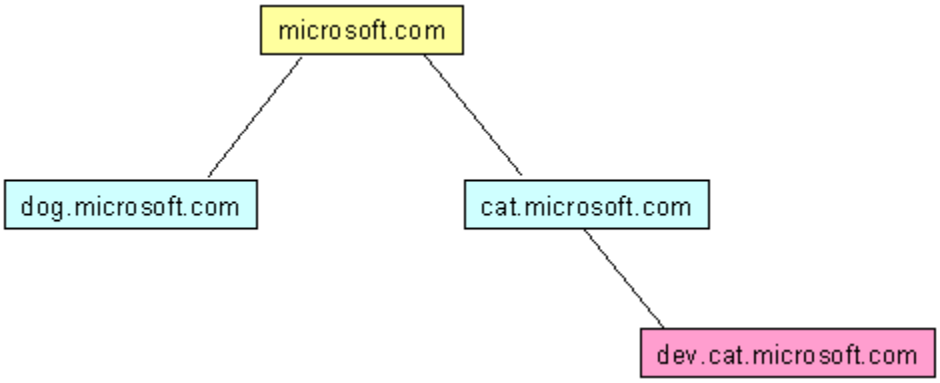
In general, the NetBIOS name will be similar to the first naming component of the DNS-style name; however, NetBIOS restricts the type and number of characters in the name and subsequently the two names may look different from one another. Upon creation of the domain, the Active Directory administrator can configure both the DNS-style and NetBIOS domain names.

In the initial release of Windows 2000, Active Directory domains cannot be renamed.

Domain Tree

Domains are represented in a parent and child hierarchy known as a domain tree.

Active Directory is made up of a number of domains that are arranged in a hierarchical fashion. Standard DNS domain names are used to represent the tree structure (for example, **dog.microsoft.com**).



A collection of domains that are in the same tree have a *contiguous namespace*. When different domain trees are installed in Active Directory, they have a *disjoint namespace*.

Forest

When a collection of domains and domain trees are joined together to form a single Active Directory, this is called a forest (or enterprise). A forest uses a single schema and configuration definition, which is replicated to all domain controllers in every domain. A forest is the *outer boundary* for Active Directory, because the directory cannot be larger than the forest itself, but non-transitive trusts can be implemented between domains that

reside in different forests (including Windows NT 4.0 domains). However, a multiple forest environment can present certain challenges to Exchange 2000 designs.

The First Domain

The first domain within an Active Directory forest plays an extremely important role. In fact, the name of the entire forest is based on the DNS name of the first domain that is installed in the forest. There are two important factors to take into consideration when defining the first domain:

- This domain can never be removed from the forest
- Other domains cannot be represented above the first domain in the domain tree; If the first domain is named **cat.microsoft.com**, it is not possible to create *microsoft.com* afterwards; however, other domain trees such as *msnbc.com* can be installed in the forest afterwards. The key to remembering this is that you can't build up, but you can build sideways

The design strategy for most large companies is to create the first domain as a *placeholder* for the rest of the forest. This domain does not contain any user or computer accounts except for two or three domain controllers. If you plan to create this domain and store it in a separate tree from the rest of the organization, you should still have the domain name registered on the Internet, and it should *not* have an arbitrary name, such as **firstdomain.com**.

Building this placeholder domain ensures that companies who have decentralized information technology departments can bring their domains online when they are ready without having to wait for other departments to catch up. This prevents departments from creating their own forests.

Domain Controller

A server that can authenticate users for a domain is called a domain controller. There must be at least one domain controller in each domain. Each domain controller holds a complete replica of the domain naming context for the domain to which it belongs, and a complete replica of the configuration and schema naming contexts for the forest. You can promote a member server to a domain controller using the Dcpromo utility. Additionally, you can demote a domain controller back to a member server using the same utility.

A domain controller supports a number of protocols including MAPI and Lightweight Directory Access Protocol (LDAP)(port 389).

Domain Mode

An Active Directory domain can be in either mixed-mode or native-mode. In mixed-mode, the domain is restricted to limitations (such as 40,000 objects) imposed by the Windows NT 4.0 domain model; however, it is possible to place a Windows 2000 domain controller in a Windows NT 4.0 domain. For this to happen, the first domain controller takes on the role of the NT 4.0 primary domain controller (PDC). As a result, mixed-vintage domain controllers operate in a single domain. While a domain is in mixed-mode, it functions like a NT 4.0 domain and has the same scaling constraints. Other Windows NT 4.0 backup domain controllers see the Windows 2000 domain controller as the PDC. When multiple Windows 2000 domain controllers are present in a mixed-mode domain, you can dictate, through the administration interface, which server functions as the PDC.

To experience all of the scalability and functionality within an Active Directory domain, you must switch the domain to *native-mode*. Switching to native-mode, which is irreversible, allows the directory to scale up to millions of objects and overcome the constraints of the legacy Security Accounts Manager (SAM), but requires that all domain controllers be upgraded to Windows 2000. A domain in native-mode allows for rich group creation and nesting, which is advantageous for Exchange 2000. A native-mode domain cannot include Windows NT 4.0 domain controllers, although Windows NT 4.0 member servers are fine. Clients do *not* need to be upgraded to belong to a native-mode domain.

1.

When multiple domains exist in the forest, you can change the mode of each domain one at a time; therefore it's not a requirement that parent or child domains be in native-mode before you switch your domain to native-mode.

Having your domains in native-mode not only provides the operating system with additional scalability, but also eases the Exchange 2000 installation process, because it is possible to create and use Universal Security groups for Public Folder access.

Site

A site is a local, logical collection of Internet Protocol (IP) subnets. All computers that are in the same site have high-speed connectivity—local area network (LAN) speeds—with one another. Unlike an Exchange 5.*x* site, an Active Directory site does not include a unit of namespace. For example, multiple sites can exist within a single domain, and conversely, a single site can span multiple domains. Synchronous Remote Procedure Calls (RPC) replicates the domain naming context within a domain. The site definition can subsequently be used to reduce RPC replication traffic by building bridges. The administrator builds *Site Links* to describe the network, while the Knowledge Consistency Checker (KCC) automatically creates connections based on the connectivity to establish an efficient, reliable replication topology.

The concept of a site in the Active Directory has been inherited from Exchange Server, although it is important to recognize that an Active Directory site only defines zones in the underlying network where high bandwidth is present. The concept of a site in the Exchange directory defined the unit of administration, namespace, and dictated message routing.

Exchange 5.x site	***Active Directory site***
Unit of namespace in the X.500 structure	No relationship with Active Directory domain structure
Unit of administration	No relationship to Active Directory administration
High-bandwidth, reliable network connections between all servers	High-bandwidth, reliable network connections between all servers
Defines single-hop message routing	Has no effect on message routing and little effect on the rest of Exchange 2000 Server

Naming Contexts

A naming context (NC) is a self-contained section of the directory hierarchy that can have its own properties, such as replication configuration and permissions structure. Active Directory uses naming contexts to define the boundaries for information held within the database structure; for example, a standard domain controller always holds three NCs in its database (Ntds.dit), which are:

- Configuration (for example, cn=Configuration,dc=cat,dc=microsoft,dc=com)
- Schema (for example, cn=Schema,cn=Configuration,dc=cat,dc=microsoft,dc=com)
- Domain (for example, dc=cat,dc=microsoft,dc=com)

In multi-domain forests, domain controllers that belong to different domains have a common configuration and schema naming context, but they have a unique domain naming context. Exchange 2000 stores the majority of its information in the configuration naming context that eases global administration, because this context is replicated forest-wide.

1.

For complete reference, Exchange Server 5.*x* uses five naming contexts; Organization, Address Book Views, Site, Configuration, and Schema.

Location Service Providers

In previous versions of Windows NT, the preferred locator service was Windows Internet Naming Service (WINS), as it provided dynamic publishing and full name to IP address mapping. In Windows 2000 and Active Directory, DNS is used as the locator service; therefore, each computer on the network is represented as a full DNS name (Pt-Server4.microsoft.com); this information is registered automatically if you deploy the Microsoft Dynamic DNS service, as it is supplied with the operating system.

Global Catalog Servers

For scalability and resilience, the local Active Directory domain is replicated between sets of domain controllers. However, some applications and services (such as Exchange 2000) require access to a full listing of the objects in Active Directory to perform forest-wide queries. Global Catalog servers exist for this purpose. Any domain controller can become a Global Catalog server. Global Catalog servers hold the configuration and schema naming contexts for the forest; a *complete replica* (read/write) of the domain naming context in which the server is installed, and a *partial replica* (read-only) of all other domains in the forest. A partial replica indicates that although every domain object is represented in the Global Catalog, only a limited number of attributes for that object are replicated to it; for example, although the user "Joe Bloggs" is represented in the Global Catalog, his telephone number is not, even though Joe's telephone number has been entered into the Active Directory.

The *Schema Manager* specifies any additional attributes that are to be replicated to each Global Catalog server. This replication policy is set and applied to all Global Catalogs in the forest. It is not possible to have different attributes replicated to different Global Catalogs in the same Active Directory.

A Global Catalog server is represented as a Service Record (SRV) in the DNS database. For example:

Node path:	**._tcp.*site*._sites._gc._msdcs.*domain***
Entry name:	**_LDAP**
Entry data:	**[0][100][3268] *server-name.domain***

By default, only the *first* server in the forest is a Global Catalog server. Even when you create additional domains, new domain controllers will only act as domain controllers. You can promote any existing domain controller to a Global Catalog through the Active Directory Sites and Services snap-in.

You can think of a Global Catalog as a super domain controller, because a Global Catalog holds all of the information for the domain and can be queried just like a domain controller. Although LDAP requests are normally passed through port 389 on a domain controller, the Global Catalog can also listen on port 3268 for Global Catalog-specific requests. Applications that are Active Directory-aware, such as Exchange 2000, can make calls specifically to this Transmission Control Protocol (TCP) port.

It is worthwhile to note that a Global Catalog server will pass back different attributes when queried depending on the TCP port used. A query to port 389 allows a client to search for objects only within the home domain of the Global Catalog, but the client can obtain all of the properties (for example, the phone number) for those objects. Conversely, a query to port 3268 allows a client to search for objects in the entire forest, although only the properties tagged for replication will be returned to the client upon request. This also applies to objects that exist in the home domain for the Global Catalog.

1.

There is no difference between a Global Catalog and domain controller for customers planning to deploy a single domain for their Active Directory. Some applications, such as Exchange 2000, must communicate with a Global Catalog to obtain information. However, in a single domain, it is possible to nominate the domain controller as a Global Catalog without incurring any server or replication overheads. This provides a local lookup facility for those applications.

Client Log On and Authentication

When a network client requires logon authentication, it uses its IP address and DNS settings, which are either static or assigned through Dynamic Host Configuration Protocol (DHCP), to locate an appropriate domain controller. When the request is received by DNS, it attempts to match the client's IP address to an Active Directory site. DNS then returns the names of the local domain controllers that can authenticate the client. The client will pick a domain controller at random and ping it before attempting to log on. If the domain controller does not respond within a reasonable amount of time, the client selects another domain controller for log on.

When a domain is in native-mode, extended functionality such as Universal Security groups can be defined. When each client logs on to the domain, the authenticating domain controller passes the client's credentials through a local Global Catalog so that it can enumerate Universal Security group access. You should be sure to allow for this additional load when determining scalability.

Per-Property Replication

When directory objects are changed in Exchange Server 5.*x*, the entire object must be replicated again throughout the organization, because Exchange only supported object-level replication. Active Directory employs a per-property replication mechanism. Not only is this more efficient, it also reduces the chances of a replication conflict because different fields in the same object can be modified in two locations at the same time. If the same field is modified, the latest update (time/date stamp) prevails. Previous versions of Exchange used the Object-version attribute (at the object level) to detect replication collisions, but Active Directory uses Property-version for each property that exists on the object.

GUIDs and Distinguished Names

As with the Exchange 5.5 directory, each object within Active Directory contains a distinguished name (DN). These are used to identify an object in the directory by a recognizable name. The DN of an object can also be used to determine its position within the Active Directory hierarchy. For example:

```
cn=pbowden,cn=Users,dc=europe,dc=microsoft,dc=com
```

Although DNs must be unique throughout the directory service, they can be changed by the administrator. This might be necessary if the directory hierarchy is reorganized because of a company restructuring.

Some applications require that an object be referred to by an identifier that is more constant. For this reason, each object also has a globally unique identifier (GUID). Once an object exists within Active Directory, a GUID is assigned to it and regardless of where the object is moved to, the GUID stays constant.

Both DNs and GUIDs can be searched and accessed programmatically by applications.

1.

Exchange 2000 Server and Active Directory

Introduction

Previous versions of Exchange utilized their own database for holding messaging system directory information, including Mailbox, Custom Recipient, Distribution List, and Public Folder directory objects. Although mailbox objects were linked to the Windows NT security database through the primary Windows NT account, no other linkage between the operating system and messaging directory existed. The main reason for this was because the NT Security Accounts Manager (SAM) was not designed to hold rich information, such as telephone numbers and certificates.

With the advent of Active Directory in Windows 2000, the directory database that the operating system provides is now capable of holding all user account details, security information, and messaging data. This means that a separate directory for Exchange 2000 is no longer necessary. Exchange 2000 is fully integrated with Active Directory.

The advantages of using one unified directory for all information, rather than several directories, are numerous and include centralized management and administration, and a single replication scheme.

Directory Objects

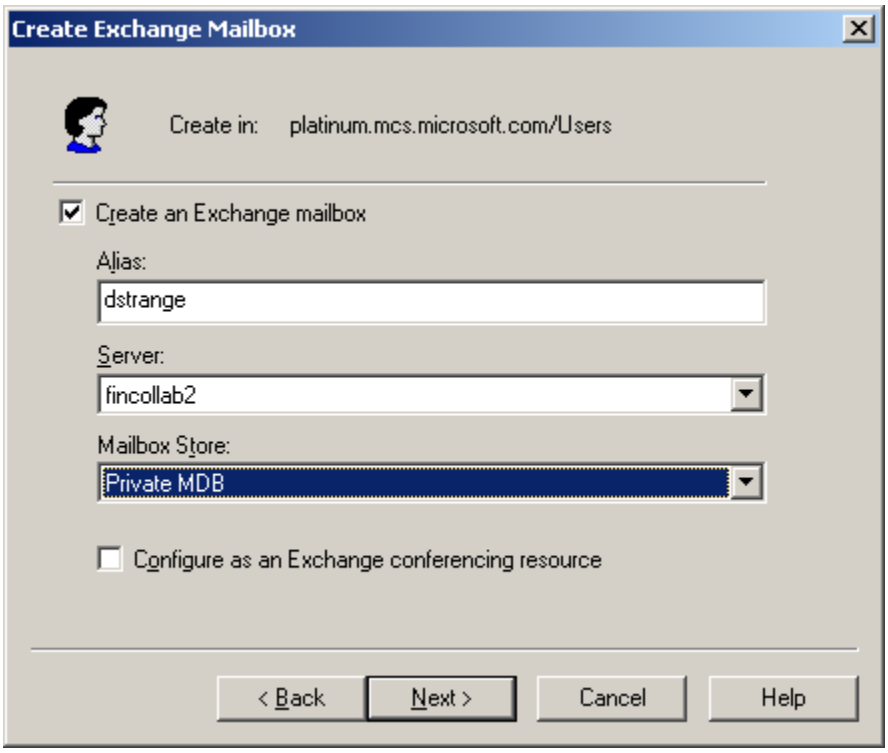
Due to consolidation and re-architecture of the directories, the object classes and terms have changed between Exchange 2000 and previous versions. The table below shows a comparison of the Exchange 5.*x* directory and Active Directory objects.

Exchange 5.x directory object	***Equivalent Active Directory object***	***Comments***
Mailbox	Mailbox-enabled user	Mailbox-enabled users are security principals in the Active Directory that can send and receive messages. Implicitly, these user types also have Simple Mail Transport Protocol (SMTP) addresses.
Custom Recipient	Mail-enabled contact	All mail-enabled contacts will have a SMTP address. Users on legacy messaging systems, such as cc:Mail and Lotus Notes, are also represented as contacts in Active Directory.
Distribution List	Mail-enabled group	Two types of groups exist in Active Directory; Security and Distribution. Additionally, the scope of the group can be set to Domain Local, Global, and Universal.
Public Folder	Public Folder	These represent mail-enabled Public Folders and can only be created through the Exchange System Manager and Active Directory Connector (ADC)—not through the standard Active Directory Users and

		Computers snap-in.

Consolidating User Management Tools

The manipulation of user objects such as mailboxes is completely coordinated within the Active Directory Users and Computers snap-in. When you install the Exchange System Manager on the Windows 2000 computer, a set of extensions are added to the standard console. This allows for, among other things, the creation of an Exchange mailbox when a new user account is defined.



Note Exchange 2000 Beta 3 does not support automatic name generation parameters for Display and Alias names.

Planning Active Directory with Exchange 2000 In Mind

Exchange 2000 servers and their users rely heavily on the following functionality delivered by Active Directory:

- Global Address List
- Rich directory access
- Groups
- Address Book Lookups for sending and routing messages
- Address Lists and Offline Address Books

1.

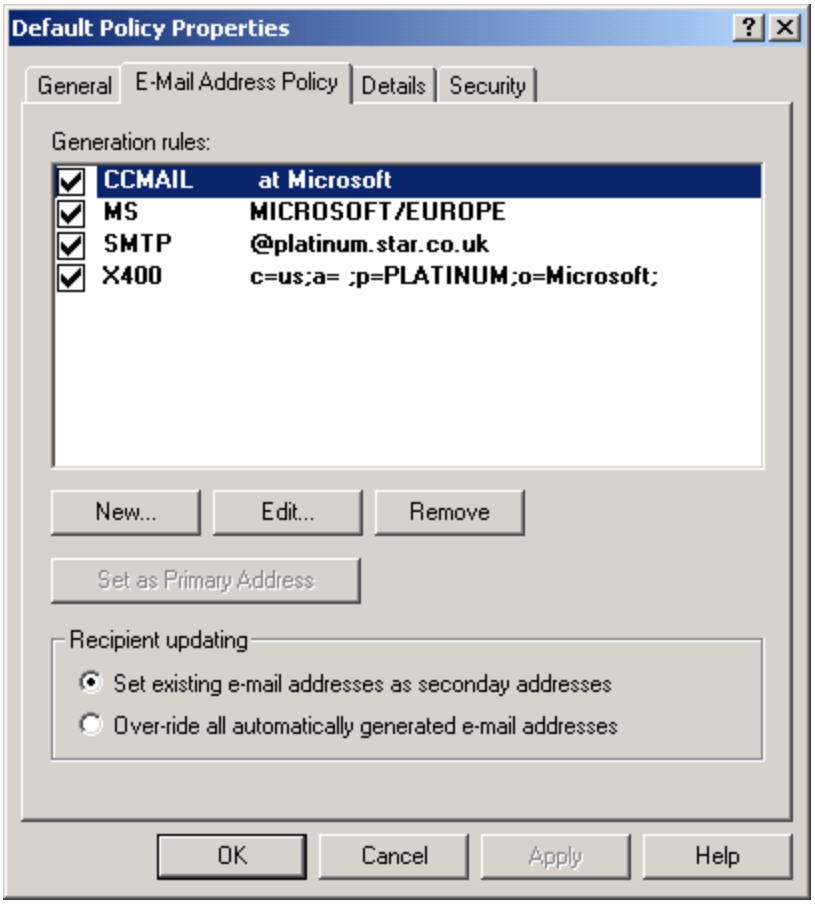
The Global Address List

Global Address List

When Outlook users want to find a person within the organization, they would normally search the Global Address List (GAL), which represents an aggregation of all messaging recipients in the enterprise. Because Exchange 2000 servers no longer host their own directory service, all data is retrieved from the Global Catalog servers in Active Directory. Because a Global Catalog server can support the Messaging Application Programming Interface (MAPI) protocol as well as Lightweight Directory Access Protocol (LDAP), Outlook clients can communicate with Active Directory using the same protocol employed by the Exchange Server 5.5 Directory Service.

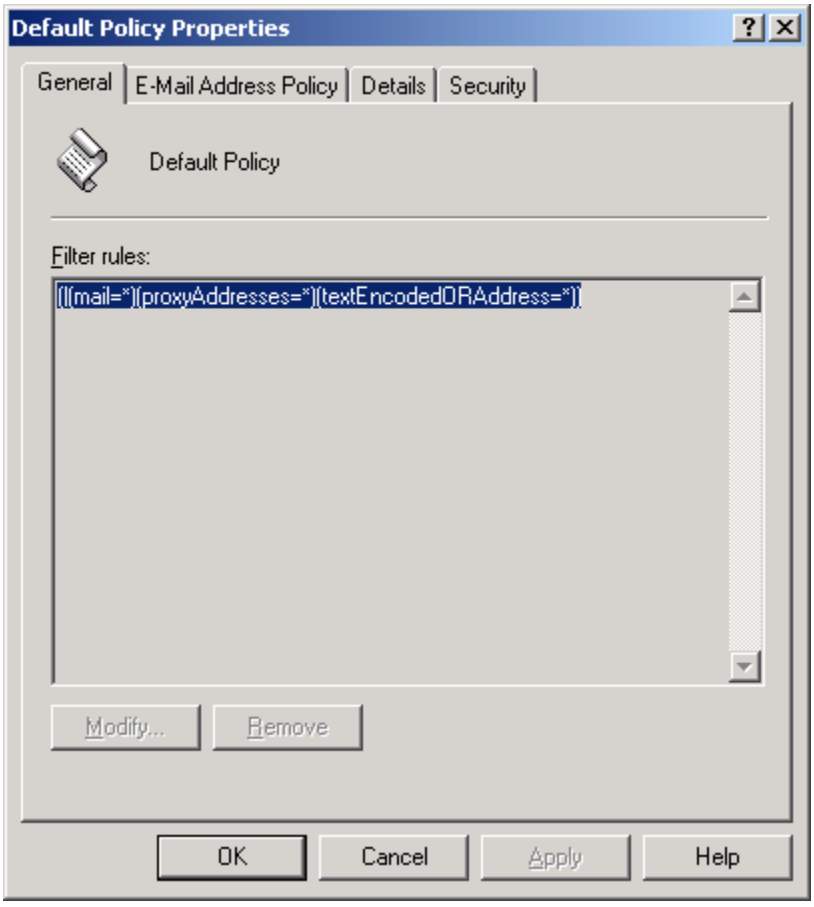
Active Directory Domain Names

Now that Windows 2000 domains use a Domain Name System (DNS) style naming convention, a domain name, such as **europe.microsoft.com**, does not dictate the Simple Mail Transport Protocol (SMTP) address for Exchange Mailbox-enabled users created within that domain. Although a user's logon name might be paul.bowden@europe.microsoft.com, the e-mail address generation is controlled by Recipient policies in the organization. In fact, a new addition to Exchange is the facility of automatically generating multiple addresses for users.



You can configure multiple Recipient policies for the organization. Each one has an RFC2254 LDAP filter rule associated with it. This allows administrators to generate proxy

addresses, such as SMTP, based on fields within the directory; for example, all research and development people in the company can have a different (or additional) SMTP address to the rest of the corporation.



Where possible, align the user's domain logon name with the SMTP address to cut down on potential end-user confusion. You may need to use a combination of Recipient policies and a User Principal Name (UPN) to achieve this; for example, consider a user named John Kenerson:

Residing domain:	redmond.microsoft.com
Recipient policy:	@microsoft.com
SMTP address:	john.kenerson@microsoft.com
Domain logon:	john.kenerson@redmond.microsoft.com
UPN logon:	john.kenerson@microsoft.com

The UPN logon is an attribute of the Active Directory account and can be set up so that John can log on to the network by a short and recognizable name, thus hiding the complexity of the underlying domain infrastructure.

Determining the Number of Global Catalog Servers Required

Exchange 2000 has a level of intelligence built into it that enables requests to be load balanced between available Global Catalog servers. When deciding on the number of Global Catalog servers required to support an Exchange 2000 deployment, consider the number of clients supported by each Global Catalog.

1.

You should plan on implementing at least two Global Catalog servers per Active Directory site for scalability and resilience. If a site spans multiple domains, it is also recommended that you configure a Global Catalog for each domain where Exchange 2000 Servers and clients have been deployed.

Deploying Active Directory Forests

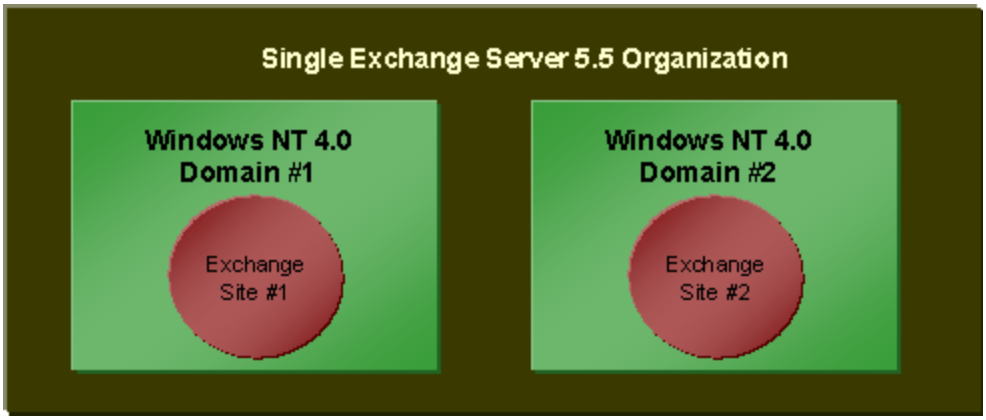
When considering the deployment strategy for your Active Directory, it is important to plan out the number of domains, the domain tree hierarchy, and the location of resources. Most companies choose to deploy a single Active Directory forest to enable a centralized management and security model. However, a company may have multiple forests for the following reasons:

- Company merger in which both companies had deployed the Active Directory
- Political boundaries between departments within the same company
- Legal situations in which directories must be completely isolated from one another
- Departmental installations
- Lack of planning and/or communication between departments

Although there are good reasons for some companies to have multiple forests within their organization, companies may end up with multiple forests by accident. If there are completely separate support structures within a company and each division must be isolated from one another, it may be reasonable to deploy multiple forests. However, it is important to remember that a single domain name can only be owned by one Active Directory. You should also remember that Active Directory supports a very granular permission and delegation structure and you do not need to deploy multiple forests to have multiple administration teams.

The worst case scenario is where a runaway department has deployed its own forest or there has been a communications failure between teams. Having multiple forests with one information technology support infrastructure can be a real challenge and can affect the deployment of Exchange 2000. In this scenario, you can create manual trusts between domains in separate forests. However, these domains are non-transitive, which means that you may end with a domain model that resembles a Windows NT 4.0 deployment, with multiple manual trusts between each domain. Additionally, the Global Catalog servers only know about the objects within their own forest; this has a direct impact on what Outlook users can see.

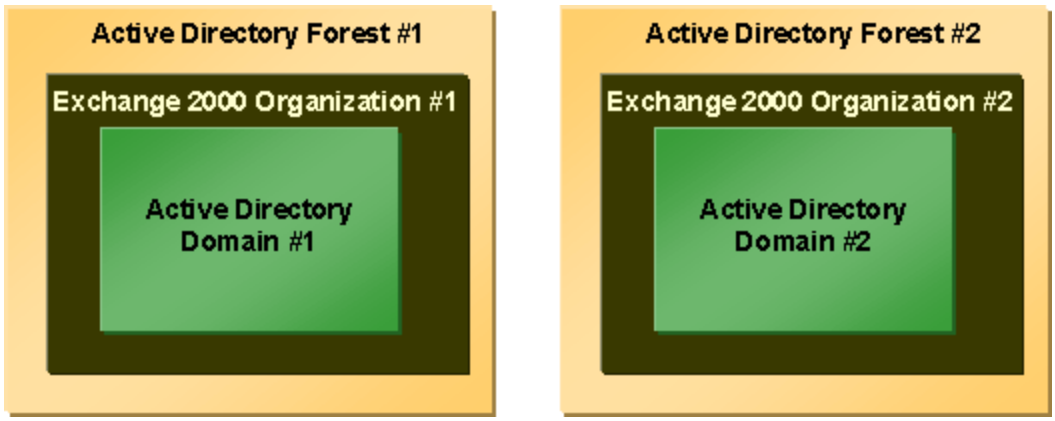
There is also a mind-set change for those people who have deployed Exchange in the past. Take, for example, two Windows NT 4.0 domains with no trusts between them. A single Exchange Server 5.5 organization with multiple sites can be deployed across these two domains without relying on the underlying security structure.



1.

Because Exchange Server 5.5 performed mail-based directory replication between sites, users in both domains were able to see all users within the same organization through one Global Address List.

However, if you consider a similar scenario with Active Directory and Exchange 2000 and it is deploy the two domains *in separate forests*, a single Exchange 2000 organization cannot span the two domains.


Active Directory Forest #1
Exchange 2000 Organization #1
Active Directory Domain #1
Active Directory Forest #2
Exchange 2000 Organization #2
Active Directory Domain #2

The ramifications of the above scenario on Exchange 2000 are as follows:

- Two separate Exchange organizations to administer
- No automatic directory replication between the two organizations, resulting in two separate Global Address Lists (users will only be able to see one of the lists)
- Cannot use routing group connectors between the two organizations (SMTP connectors and/or X400 connectors need to be used instead)
- No link state data transfer for Exchange 2000 because routing group connectors cannot be used

If the two Active Directory domains belong to the same forest, a single Exchange 2000 organization can be formed.

Synchronizing Directory Data Between Forests

If Outlook users and Exchange 2000 servers are deployed in multiple forests, you may need to replicate certain data between the two systems to achieve the functionality you need.

Directory Information

A number of different options are available for synchronizing multiple forests together. The **DirSync Control** is a standard way for synchronizing directory data from different LDAP providers, and has been submitted as an open standard to the Internet Engineering Task Force (IETF). Third-party products, such as the Compaq LDAP Synchronization Utility (LDSU), ISOCOR's MetaConnect, ZoomIt's VIA, or Siemens' DirX, can also be used.

It is important to understand that any such synchronization of directory data between forests will change the target object class. For example, a Mailbox-enabled user in Organization #1 will appear as a mail-enabled contact in Organization #2.

Public Folders

Provided with Exchange 2000 is the Public Folder Inter-organization Replication tool, which can synchronize Public Folders between different organizations. However,

replicating Public Folders between different organizations creates additional administration overhead compared to replicating Public Folders in a single organization.

Calendaring and Scheduling

The Public Folder Inter-organization Replication tool can also replicate information such as Free/Busy System folders. This allows users from different forests to schedule meetings with one another and look up free and busy time-slots.

Be aware that there is no automatic solution to for replicating user’s calendars between organizations, and subsequently users do not have the ability to open calendars across the organization’s boundaries.

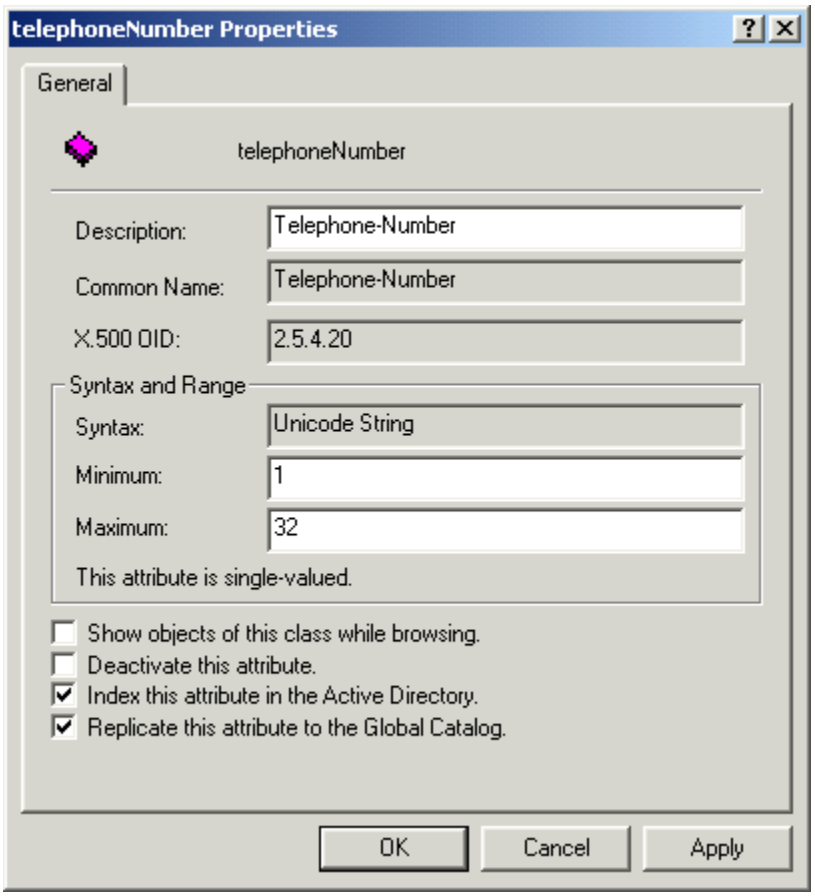
1.

Understanding Selectable Field Replication

With existing versions of Exchange and Outlook, users within an organization rely on the directory data that is populated to the Exchange directory. For example, it is not uncommon for users to look up each other's telephone numbers through this mechanism.

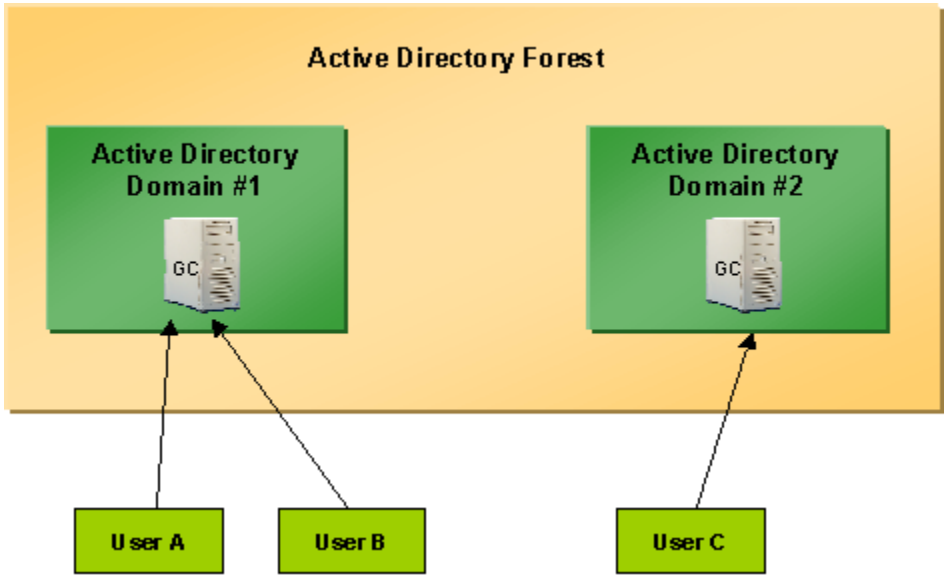
With Exchange Server versions 4.0 and 5.0, any directory data present in one Exchange server must be replicated to every other Exchange server within the organization. To reduce network traffic, Exchange Server version 5.5 implemented the notion of selectable field replication, which prevents certain attributes (but not objects) from replicating between sites. Many companies do not take advantage of this new feature because users rely on the directory information on a daily basis.

Active Directory has a similar feature in which selective attributes can be tagged for replication to the Global Catalog server. The main difference between this feature in Active Directory and previous versions of Exchange is that by default, not all attributes are tagged for Global Catalog replication. Some attributes having to be tagged manually through the Windows 2000 Schema Manager snap-in. You can enable the snap-in by typing **Regsvr32 schmmgmt.dll** at the command prompt. Once enabled, launch the Microsoft Management Console (MMC) process and add the Active Directory Schema snap-in to the console.



Because a Global Catalog holds a complete domain replica of its home domain and a partial domain replica of every other domain in the forest, Outlook users can always see

full directory details for other users within the local domain, but depending on the configuration they may not be able to see the same data for users in other domains.



In the previous example, all directory attributes have been populated into Active Directory domains, and the **Department** attribute has *not* been tagged for Global Catalog replication. Because the Global Catalog holds a complete replica of Domain #1, User B can see the department name for User A., but Users A and B cannot see User C's department name, and User C cannot see the department name for Users A and B.

Selecting Attributes to Replicate to the Global Catalog

Selecting which attributes to replicate to the Global Catalog requires careful planning. You don't want to remove any functionality that the Outlook users already have if Exchange is currently deployed, but you must consider the ramifications for replication traffic if you tag too many additional attributes.

If very slow networks are involved, you may want to survey Outlook users to find out which directory attributes they rely on. You also need to find out whether any custom Collaboration Data Objects (CDO) and/or Active Directory Services Interface (ADSI) applications require certain directory data. For example, a workflow application may require access to a Custom Attribute that holds a manager's sign-off limit.

When Microsoft Exchange 2000 Server is installed into the Active Directory, it will automatically tag many of the attributes that users normally require; however, there are some other important attributes, such as **Department**, which are not tagged automatically.

Each additional attribute that you manually tag for replication will incur roughly an additional 100 bytes of replication data per object. However, the replication traffic caused by an existing Exchange 5.*x* network will be far greater than the replication traffic that Active Directory will produce. This is based on the following assumptions:

- Each Exchange 5.*x* server in the organization must hold a full copy of the Exchange directory, but Active Directory only replicates to domain controllers and Global Catalog servers.
- Any change to an Exchange 5.*x* object causes the entire object to replicated again to the rest of the Exchange organization, which incurs roughly 5 kilobytes (KB) of replication data intra-site and 1 KB inter-site, but Active Directory uses per-property replication and the replication payload is much smaller.

1.

- Not only is the Exchange 5.*x* directory replicated around the company network, but replication traffic is also produced by Windows NT4.0 domain controllers. Active Directory consolidates these two directories into one.

Exchange 2000 Installation and Active Directory Schema

Many Lightweight Directory Access Protocol (LDAP) Directory Interface Format (LDIF) files are imported as part of the installation process for the first Exchange 2000 server in Active Directory. This process can take an extended period of time to complete. The Active Directory schema is extended with new attributes that all start with *ms-Exch.…*. Simultaneously, existing Active Directory attributes are modified, some of which affect what Outlook users can see. The table below lists the common attributes and LDAP names for a Mailbox-enabled user object. The table illustrates how these attributes differ between a standard installation of Active Directory, and an Active Directory that is enabled for Exchange 2000. The *In Global Catalog* column indicates whether the attribute has been tagged for Global Catalog server replication. The table may be helpful for planning purposes.

Attribute	***LDAP***	***Standard Active Directory***		***After Exchange 2000 Installation***	
		Index	**In Global Catalog**	**Indexed**	**In Global Catalog**
First name	**givenName**	Yes	No	Yes	Yes
Initials	**initals**	No	No	No	No
Last name	**Sn**	Yes	No	Yes	Yes
Display name	**displayName**	Yes	No	Yes	Yes
Alias	**mailNickname**	N/A	N/A	Yes	Yes
Mailing address	**streetAddress**	No	No	No	No
City	**L**	Yes	No	Yes	Yes
State	**St**	No	No	No	Yes
ZIP code	**postalCode**	No	No	No	No
Country	**C**	No	No	No	Yes
Job title	**Title**	No	No	No	Yes
Company	**company**	No	No	No	Yes
Department	**department**	No	No	No	No
Office	**physicalDeliveryOfficeName**	Yes	No	Yes	Yes
Telephone	**telephoneNumber**	No	No	No	Yes
Fax	**facsimileTelephoneNumber**	No	No	No	No
Home telephone	**homePhone**	No	No	No	No

1.

Attribute	*LDAP*	*Standard Active Directory*		*After Exchange 2000 Installation*	
		Index	**In Global Catalog**	**Indexed**	**In Global Catalog**
Manager	**manager**	No	Yes	No	Yes
SMTP Address	**Mail**	Yes	No	Yes	Yes
Custom attributes (all)	**extensionAttribute-xx**	N/A	N/A	No	No

Making Schema Changes

As the needs of an organization change, the Active Directory schema can develop and grow to accommodate those changes. Be aware that some operations in Active Directory can affect the infrastructure and replication mechanism. For example, if you deploy Active Directory and then decide to tag another attribute for Global Catalog replication, you can make the change quite easily in the Schema Manager. However, this will cause all Global Catalogs to set their Update Sequence Numbers (USNs) to 0. As a result, all objects (not just the changed property) in Active Directory must be replicated to each Global Catalog again, which causes a significant network overhead.

Additionally, if you install an application (such as Exchange 2000) that tags attributes for replication to the Global Catalog, it will also have the same impact on Active Directory. For this reason, it is important to do as much planning up-front as possible to mitigate this issue.

For companies planning to deploy Active Directory and install Exchange 2000 servers at a future date, it is advantageous to import the Exchange-specific schema changes before Active Directory becomes too large. This can be accomplished by one of the following methods:

- Run the Exchange 2000 SETUP program with the **/SchemaOnly** command-line switch (**Setup /SchemaOnly)**
- Manually import the LDIF files on the Exchange 2000 installation CD

1.

Implementing Groups in Active Directory

Exchange Server 5.5 Distribution Lists

Distribution Lists in Exchange Server 5.5 serve two purposes; for sending mail to large groups of recipients and for assigning permissions to Public Folders. A single object class is used for a Distribution List. Because of the directory architecture in previous versions of Exchange, Distribution List membership is replicated to each Exchange server in the organization. By default, all users are able to view the membership for a given list, although this can be restricted on a per-list basis. It is possible for any Exchange message transfer agent (MTA) to expand the contents of a Distribution List because each list is represented on every server.

Active Directory Groups

In Active Directory, the equivalent of a Distribution List is a *group*. The following table describes different types of groups.

Group type	***Explanation***
Security	These are *security principals.* Members can be granted access to network resources.
Distribution	These are *non-security principals*. Members cannot be granted access to network resources. Distribution groups exist for sending bulk e-mail.
Group scope	***Explanation***
Domain Local	Can contain user accounts, Global groups, and Universal groups from any domain. Can also contain Domain Local groups from the same domain, if the domain is in native-mode. Can contain user accounts and Global groups from any domain, when created in a mixed-mode domain. Can only be put in other Domain Local groups and permission lists in the same domain. Can be converted to a Universal group in a native-mode domain, if it does not contain another Domain Local group. The group object is present in the Global Catalog, although the membership is not published to it.
Global	Can contain user accounts and Global groups from the same domain, if the domain is in native-mode. Can contain user accounts from the same domain, if created in a mixed-mode domain. Can be put in groups and permission lists in any domain. Can be converted to a Universal group in a native-mode domain, if it is not a member of any other Global group. The group object is present in the Global Catalog, although the membership is not published to it.
Universal	Can contain user accounts, Global groups, and Universal groups from any domain if the domain is in native-mode

	Universal groups of type Security can only be used in native-mode domains. Universal groups of type Distribution can be used in mixed-mode and native-mode domains. Can be put in Groups and permission lists in any domain. Cannot be converted to any other group scope.

When the administrator attempts to create a new group in a mixed-mode domain, by default it is configured as a Security group with a Domain Local scope. A new group in a native-mode domain defaults to a Security group with a Universal scope. In mixed-mode domains, you cannot change a group's scope after you have created it.

Recommendations for Implementing Groups

The type and scope of the group that you use for Exchange 2000 depends on the business and user requirements. For full flexibility, implement Universal groups of type Security. Although this is a Security group definition, it can be mail-enabled (by adding a Simple Mail Transport Protocol (SMTP) address) and viewed in the Global Address List. However, a disadvantage of Universal groups of type Security is that they can only be created in native-mode domains. Fortunately, to move from a mixed-mode to native-mode domain, you only need to upgrade the domain controllers to Windows 2000. Moving to native-mode domains eases the upgrade and deployment process for Exchange 2000 and provides additional directory scalability.

Another point to consider when deploying Universal groups is that their membership is published to the Global Catalog servers, so any membership change causes replication traffic. Although Active Directory supports property-level replication, the membership for a group is held in a multi-valued property on the group object. Therefore, if the group is large, a significant amount of replication traffic can be generated. To mitigate this risk, place user objects in other Universal groups and then nest these groups under an umbrella Universal group. When the membership changes for a user in the group, the large Universal group object is not changed and no replication traffic is created. As Universal groups have been used in this scenario, Outlook users can still view full membership of both the umbrella group and its subgroups. You can use Global groups instead of Universal groups, however, if you use this method of configuration, Global groups do not have their membership directly published to the Global Catalog servers and this may impact the ability of an Outlook client to view membership at the recipient level.

When a message is sent to a group, the SMTP service must expand the membership of the group object. If it is a Domain Local or Global group defined in the local domain, the membership list can be retrieved from any local domain controller. Additionally, if it is a Universal group and users appear directly on the list, the membership can be obtained from any local Global Catalog server.

If a message is sent to a Domain Local or Global group that has been created in another domain, or if a Universal group contains Global groups that are in other domains, there are two choices for expansion:

- Forward the message to the remote domain and let it expand there.
- Expand the message locally, but make direct LDAP calls to a domain controller in the remote domain to retrieve the membership. This implies that there is direct Internet Protocol (IP) connectivity between the expanding server and the remote domain controller.

The disadvantage of the second method is that, depending on the speed of the network, remote retrieval may take some time, which can slow down message delivery. If an Exchange server exists in the remote domain, it may be more efficient to set the expansion to that server instead of remotely retrieving the membership.

1.

You must decide whether to create a Universal group for the list or whether to use Domain Local or Global groups and accept the membership retrieval for remote domains whenever the list needs to be expanded. Consider the following questions when deciding on the group type:

- Are you using single or multiple Active Directory domains within the organization?
- Is direct IP connectivity possible between all domains?
- How many membership changes will occur within a given time period; for example, a week or a month?
- Who will be sending the majority of mail to the list: users in local domains or users in remote domains?

Also keep in mind, when implementing Domain Local and/or Global groups, Outlook users will not be able to view the membership of the group unless the Domain Local or Global groups are also defined within the same domain as the user.

You must also remember that groups are used for determining Public Folder access. Unlike previous versions of Exchange, the store does not need to expand a group when a user accesses a Public Folder. Because all Public Folders' access control lists (ACLs) are based in Active Directory, group membership is carried in a users access token and is presented to the Exchange server upon connection to the resource.

You should always keep the number of members in a group (of any type) below 5000. If a group needs to contain more than 5000 users, nesting should be used. In fact, for efficiency and scalability reasons, you should consider nesting groups that contain more than 500 members.

The following list will help you decide on the most appropriate way to implement your groups.

Type: Security groups

Advantages

- Can be mail-enabled by assigning an SMTP address, allowing the group to act as a Distribution List equivalent.
- Can be used for assigning permissions to Public Folders in Exchange 2000.
- Can be useful if you want to also assign network permissions for the members in this group.
- Can lower the number of groups and maintenance in Active Directory and the messaging system, because these can act as pseudo-Distribution Groups.

Disadvantages

- Accidentally placing users in the group can give them unauthorized access to network resources.

Type: Distribution groups

Advantages

- Can be used for bulk mailing.
- Can be used as Universal groups even in a mixed-mode domain.

Disadvantages

- Permissions cannot be assigned to network resources.
- Permissions cannot be assigned to Public Folders in Exchange 2000 Beta 3.

1.

Scope: Domain Local groups

Advantages

- Membership is not published to the Global Catalog server, so membership changes do not incur Global Catalog replication.
- Outlook clients can view full user membership if they are located in the domain in which the group exists.
- Membership can consist of users from any domain.

Disadvantages

- Permissions cannot be assigned to network resources and Public Folders in other domains.
- Outlook users in other domains cannot view the full membership.
- Group membership must be retrieved on demand if expansion takes place in a remote domain.

Scope: Global groups

Advantages

- Membership is not published to the Global Catalog server, so membership changes do not incur Global Catalog replication.
- Outlook clients can view full user membership if they are located in the domain in which the group exists.
- Permissions can be assigned to network resources and Public Folders in any domain.

Disadvantages

- Group can only contain recipient objects from the same domain.
- Outlook users in other domains cannot view the full membership.
- Group membership must be retrieved on demand if expansion takes place in a remote domain

Scope: Universal groups

Advantages

- Membership can consist of any object in the forest
- Outlook users in any domain can view full membership.
- Membership never has to be retrieved from remote domain controllers.
- Group can be used in mixed-mode domains if type is set to Distribution.

Disadvantages

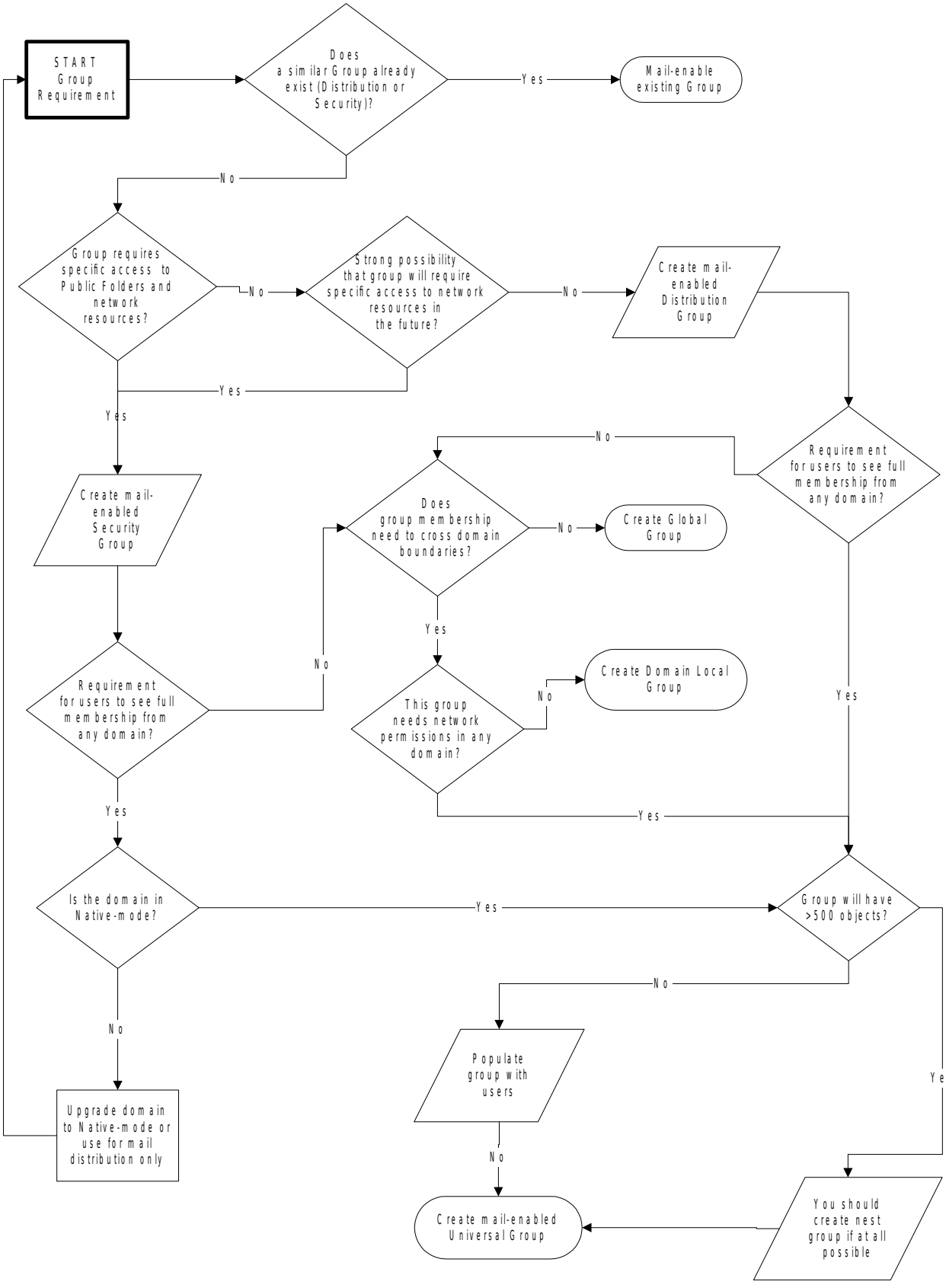
- Membership modifications incur replication to the Global Catalog servers.
- Universal Security groups can only be created if a domain is in native-mode.

Group Creation Decision Process

Implementing the correct group type for Exchange 2000 Beta 3 mailing and Public Folder access is very important. Before you create the group, a full understanding of how it will

be used is recommended. The scope, size, and membership of the group are key factors when implementing different group types. Use the following diagram to help guide you through your decision process.

1.


START Group Requirement
Does a similar Group already exist (Distribution or Security)?
Yes
Mail-enable existing Group
No
Group requires specific access to Public Folders and network resources?
No
Strong possibility that group will require specific access to network resources in the future?
No
Create mail-enabled Distribution Group
Yes
Yes
Create mail-enabled Security Group
Requirement for users to see full membership from any domain?
No
Does group membership need to cross domain boundaries?
No
Create Global Group
Yes
Requirement for users to see full membership from any domain?
No
This group needs network permissions in any domain?
No
Create Domain Local Group
Yes
Yes
Yes
Is the domain in Native-mode?
Yes
Group will have >500 objects?
No
Yes
No
Upgrade domain to Native-mode or use for mail distribution only
Populate group with users
No
Create mail-enabled Universal Group
You should create nest group if at all possible

Requirement: To allow the sales team in Toronto to e-mail each other with lead information.

Solution: Create a group with type Distribution and scope Global.

Reasoning: If all of these members are in the same domain, you can use a Global group. The purpose of the group is to enable members to send e-mail to each other, so access to network resources are not a consideration. It may be appropriate to create a Security group now, so that the team can be given access to network resources and printers. This will prevent having two groups with the same membership if requirements change in the future. The list will only be used by members of the team in the same domain, so membership will not need to be retrieved from remote domain controllers.

Requirement: The Worldwide Product Marketing Unit wants to have a list created so that anyone within the company can e-mail them. Users do not need to know who is on the mailing list.

Solution: Create a group with type Distribution and scope Domain Local.

Reasoning: You cannot use Global groups, because the members of the group are in a multitude of domains. Because network access is not desired and the membership does not need to be published to the Global Catalog, you do not need to create a Universal group. The Marketing Unit also sees frequent changes to their group; therefore, it is not appropriate to create a Universal group. However, to prevent membership from being remotely retrieved when a user in a remote domain sends a message to the group, the list is set to expand on a server in the domain in which the group is defined.

Requirement: To create a small mail-based discussion forum for a new company product and to give the team access to all of the Web sites and network shares that contain information about the product.

Solution: Create a group with type Security and scope Universal.

Reasoning: Members of the discussion forum can be located anywhere in the world. As a result, there is a strong possibility that members are in different domains; hence, the Universal group. The team is quite small and it is not envisaged that membership will change on a frequent basis, so it is appropriate to put the user membership directly within the Universal group. This increases usability because members can see who is on the list before they send sensitive or confidential information. The group created is of type Security because network resource access is also a requirement. Note that the domain in which the Universal group is created must be in native-mode.

Requirement: To create a mailing list to bulk mail security announcements and important information to the entire company.

Solution: Create a group with type Distribution and scope Global in every user domain, and then nest these Global groups into a Universal group

Reasoning: Access to network resources is usually controlled on a per-team basis or through the default access control entry, which grants permissions for users who have not been explicitly defined on the access control list. As a result, it is extremely unlikely that you will need to grant permissions on network resources for these groups. Membership for this group will change fairly frequently, making it inappropriate to populate full membership into a single Universal group. When a membership change takes place, only the domain controllers within that domain need to be updated with the modification. Global Catalog servers will not have to replicate any data. From a user's viewpoint it is extremely unlikely that they will want to see the membership of this group.

The Global groups are nested within the Universal group, so it is very easy for users to e-mail the entire company using a single address (permissions permitting) instead of selecting each individual team or region. Another advantage of this method is that users can send e-mail to an individual group very easily; for example, notifications about the Euro currency to the European group.

Domain Local groups cannot be used in this scenario because they cannot be placed into Universal groups.

Coexisting with and Upgrading from Previous Versions of Exchange

In a coexistence scenario, the Active Directory Connector (ADC) replicates Exchange Distribution Lists to Active Directory groups. In Exchange 2000 Beta 3, Exchange Server 5.*x* Distribution Lists replicate to the Active Directory as a Universal Distribution group.

If the ADC has not been configured, the Exchange 2000 Beta 3 server upgrade process converts any existing Distribution Lists in Exchange Server 5.5 to Universal groups of type Distribution if the Active Directory domain is in mixed-mode, and Universal groups of type Security if the Active Directory domain is in native-mode.

1.

Address Book Lookups

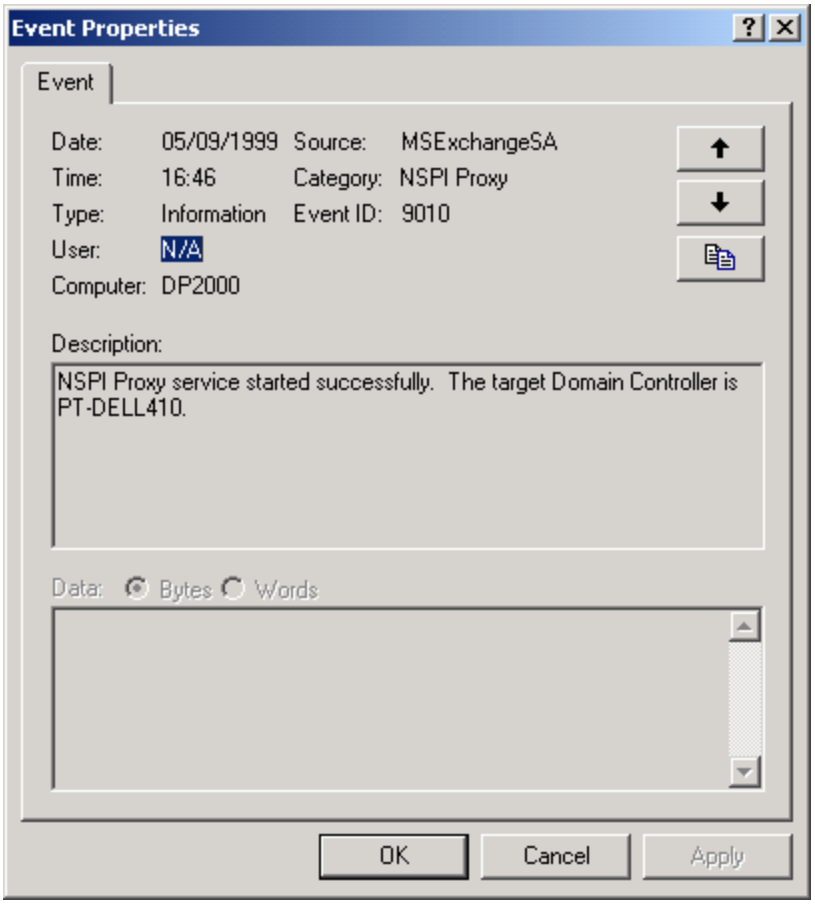
You need to be able to gauge how an Outlook user and a Microsoft Exchange 2000 server access Active Directory. In previous versions of Exchange, every server had a copy of the directory, Outlook users referred to the directory that housed their Mailbox, and the message transfer agent (MTA) would always use the local directory to route messages. Now that Exchange 2000 no longer has a directory of its own, it has to make calls into Active Directory.

DSProxy

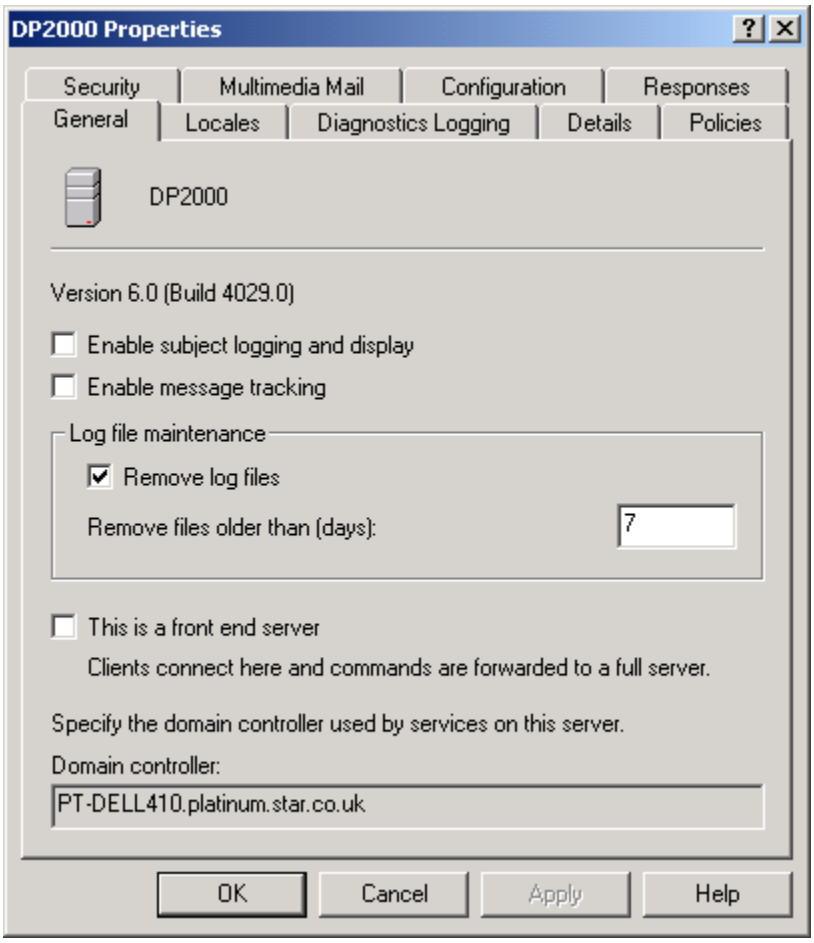
The DSProxy process acts as a facilitator to allow Outlook clients to access the data within Active Directory. It performs two major functions:

- Proxying directory requests on behalf of clients to Active Directory through the Name Service Provider Interface (NSPI)
- Referring *smart* messaging application programming interface (MAPI) clients directly to the Active Directory (RFR)

Upon startup, the Exchange System Attendant finds the most appropriate Active Directory server in the domain through the Domain Name System (DNS) resolver and will pass its name through to the DSProxy process (DSPROXY.DLL). This is signified by event 9010 for the MSExchangeSA process.



It is also possible to ascertain which Active Directory domain controller a given Exchange server is using by viewing the properties of the Exchange computer in the Exchange System Manager snap-in.



In some situations, the administrator should explicitly set the server that DSProxy uses. You can accomplish this by using the following registry parameters:

HKEY_LOCAL_MACHINE \ System \ CurrentControlSet \ Services \ MSExchangeSA \ Parameters

Value name: NSPI Target Server
Value type: STRING
Value data: *GC-server-name*

HKEY_LOCAL_MACHINE \ System \ CurrentControlSet \ Services \ MSExchangeSA \ Parameters

Value name: RFR Target Server
Value type: STRING
Value data: *GC-server-name*

The DSProxy NSPI works by *blindly* forwarding MAPI directory system (DS) requests to a Global Catalog server. This means that the remote procedure call (RPC) packet is not opened and evaluated because this would incur a significant overhead on the Exchange 2000 server and complicate the security structure. The process begins by creating a

1.

listening thread for each supported network protocol and a single working thread for each processor. This can accommodate up to 512 client connections and dynamically adds more threads as required. A socket mapping table keeps a reference of connections between clients and servers, ensuring that the correct responses from Active Directory are passed to the associated client.

If the server being used by DSProxy fails, DSProxy issues a call-back to the System Attendant (Mad.exe), which in-turn issues DSProxy a new server name to use. This is known as re-targetting. The System Attendant will not re-target DSProxy without receiving a request for this operation. Note that in Exchange 2000 Server Beta 3, if the server being used by DSProxy fails, the clients accessing that Exchange server must be restarted to make use of an alternate Active Directory server.

DSProxy works over Transmission Control Protocol/Internet Protocol (TCP/IP), Internetwork Packet Exchange (IPX), and AppleTalk protocols; however, does not work over network basic input/output system (NetBIOS).

Client Access to Active Directory

Exchange Client and Outlook 97/98

Older MAPI clients, such as Exchange and Outlook 97/98, make MAPI Directory Service (MAPI DS) requests to an Exchange server. These requests come in different forms and can range from resolving a fragment of text typed on the **To:** line of a message into a user name to showing objects from the Global Address List. To make Exchange 2000 compatible with the existing MAPI client base, an Exchange 2000 server proxy any MAPI DS requests through to a local Global Gatalog server on the network. It is the DSProxy process on the Exchange 2000 server that accomplished this task. The proxying process is simply a forward of the packet; it does not change the request into (Lightweight Directory Access Protocol) LDAP. Active Directory supports a number of protocols, including LDAP and MAPI DS, so an Outlook directory request is completely valid, even directly against an Active Directory server.

The Global Catalog server returns the result to the Exchange 2000 server, and then proxy the result on to the MAPI client. This process is seamless to the user and takes no extra time to complete.

Traffic and Load Generated Through the DSProxy Process

The following steps describe the communication process that takes place for a one-recipient name lookup:

1. The MAPI client sends one network packet to the Exchange 2000 server. The packet contains the name to look up in plain text.
2. The Exchange 2000 server proxies the request and sends it to a local Global Catalog.
3. The local Global Catalog returns the result to the Exchange 2000 server.
4. The Exchange 2000 server returns the result to the MAPI client.
5. The MAPI client returns an acknowledgement to the Exchange 2000 server.
6. The Exchange 2000 server proxies the acknowledgement to the local Global Catalog.

The directory lookup only produces six frames on the network. Even when multiple names need to be looked up in the directory, the name fragments are sent in the one request packet.

1.

If the user chooses to browse the Global Address List, the same process takes place. A few extra frames are seen on the network as the user scrolls through the Address List, thus causing the client to retrieve more information.

Outlook 2000

Newer versions of the Outlook client, such as Outlook 2000, use a slightly different method for address book access. Initially, Outlook expects to find the directory service on the home Exchange server. Because the version of the Exchange server being used on the system can be determined only after loading EMSMDB32.DLL (which is after the address book provider—EMSABP32.DLL), Outlook will go through the DSProxy process for the first session. Once the client has contacted the DSProxy service (it will try all available transport protocols), a special referral will be passed back to the client informing it that all future directory requests should be sent to the Global Catalog server. Outlook will persist this referral in the MAPI profile under the following key:

HKEY_CURRENT_USER \ Software \ Microsoft \ Windows NT \ CurrentVersion \ Windows Messaging Subsystem \ Profiles \ *profile name* \ dca740…2fe182

Value name: 001e6602
Value type: String
Value data: *DirectoryServer.domain*

(Example: \\TABBY.cat.microsoft.com)

The referral mechanism reduces the load on the Exchange 2000 server and the latency for address book lookups; however, when an explicit server name is entered into the profile, Outlook will need to be restarted if that Active Directory server fails. In this event, the Exchange 2000 server will pass Outlook a new referral to use.

In some scenarios it is desirable to require Outlook clients, even the latest versions, to go through the DSProxy process without being referred. Probably the best example of this is when a firewall exists between the client computers and Active Directory servers. The firewall can be opened up to allow the Exchange 2000 server to access Active Directory.

This can be achieved by instructing the Exchange 2000 server so that it does not give referrals and is implemented through the following registry parameter:

HKEY_LOCAL_MACHINE \ System \ CurrentControlSet \ Services \ MSExchangeSA \ Parameters

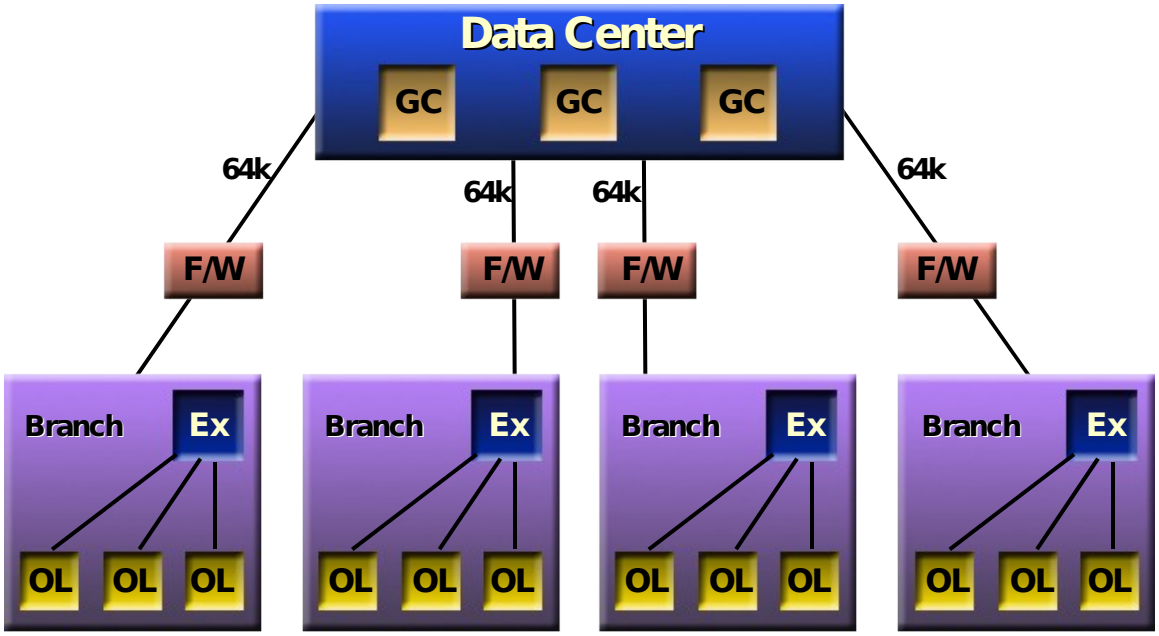
Value name: No RFR Service
Value type: DWORD
Value data: 0x1

Exchange 2000 Directory Access Cache

In many organizations the Global Catalog server is a precious resource and should not be monopolized by a single application. Additionally, there may only be one or two Global Catalog servers in a physical location, making the monitoring of its workload important. To reduce the burden on the Global Catalog, each Exchange 2000 server has a Directory Access Cache *(*DSAccess**)**. This allows the store and other processes such as the MTA to perform directory lookups and for those to be cached for a period of time without sending the same query to the Global Catalog again. All directory access, except Address Book lookups from MAPI clients and certain portions of Simple Mail Transport Protocol (SMTP) inbound/outbound routing, will go through the DSAccess process. This can greatly increase both the level of performance of the network and Active Directory servers.

By default, a maximum of 4 megabytes (MB) of directory entries are cached for a period of 10 minutes. These parameters can be changed through the registry of the Exchange 2000 server. You can monitor the effectiveness of the cache through the Windows 2000 Performance Monitor.

1.



In certain environments, such as retail banks, this facility will be invaluable for reducing wide area network (WAN) traffic. Consider the following scenario:

Here we see that each branch has its own Exchange 2000 collaboration server, and this grants users fast access to their data. For a company that potentially has thousands of branches, it is unworkable to place a Global Catalog server in every location because this would produce too much replication traffic on the network. Many retail organizations have low speed links between the branch offices and the data center: 64 kilobytes (KB) per second is fairly typical. There may also be a firewall in place between the branches and the data center, because it is likely that ATM traffic is also being sent across the same network line.

Exchange 2000 is easy to deploy in this scenario because directory requests can be proxied on demand between the Exchange 2000 server and the collection of Global Catalogs at the data center. With Exchange 2000, the global directory does not need to be replicated to every server as it did with previous versions of Exchange.

Address Book lookups from MAPI clients are either be proxied or referred by the DSProxy process, depending on the version of the software installed on the client. In this scenario, because the firewalls are locked down between the branches and the data center, all Outlook clients can be required to go through the DSProxy process by implementing the **No RFR** registry value (see previous section).

Manually Adjusting the Cache Parameters

In some environments it is useful to disable the Directory Access cache, and this may also be necessary for troubleshooting directory problems. The following registry value dictates whether the cache is active:

HKEY_LOCAL_MACHINE \ System \ CurrentControlSet \ Services \ MSExchangeDSAccess

Value name: CachingEnabled
Value type: *REG_DWORD*
Value data: 0x1 (Enabled – Default) 0x2 (Disabled)

1.

When tuning a caching mechanism for a particular environment, it is important that you get the right balance between performance and data access; for example, in the retail bank scenario described previously, it would be useful to increase the amount of time entries can exist within the cache to further reduce network traffic. The balance is that the cache may become old if entries remain there for too long. A further issue is that increasing the total time entries remain in the cache also increases the amount of system memory the cache uses. You must carefully consider the environment before making adjustments and you should thoroughly test and document any changes. You can adjust the expiry time for cached entries using the following registry value:

HKEY_LOCAL_MACHINE \ System \ CurrentControlSet \ Services \ MSExchangeDSAccess \ Instance0

Value name: CacheTTL
Value type: *REG_DWORD*
Value data: 0x600 (seconds)

There are two ways you can control the size of the cache; by limiting the number of entries that can be cached, or by limiting the amount of memory that the cache process consumes. Each cached entry consumes roughly 3.6 KB of memory and the cache process itself carries an overheard of approximately 2.5 MB. In most environments, you can leave the cache ceiling at its default settings, although in some cases you can tune the cache parameters. For example, you might want to adjust the cache parameters if you are a service provider who needs high performance. In general, it is better to limit the size of the cache process (*MaxMemory*) rather than limit the number of entries (*MaxEntries*), because this allows for better control of memory usage.

HKEY_LOCAL_MACHINE \ System \ CurrentControlSet \ Services \ MSExchangeDSAccess \ Instance0

Value name: MaxMemory
Value type: *REG_DWORD*
Value data: 0x0 (Kilobytes – default = 4096)

Value name: MaxEntries
Value type: *REG_DWORD*
Value data: 0x0 (number of entries – default = 0 (unlimited))

Getting Directory Data

Depending on the type of request being made, an Exchange 2000 server may go to different Active Directory servers to retrieve the data. An Exchange 2000 server usually establishes a number of LDAP connections to domain controllers and Global Catalog servers that are close by. There are two main directory activities that an Exchange 2000 server performs; Address Book and configuration data lookups. For an Address Book lookup, the Exchange 2000 server queries the closest Global Catalog server available to it.

When an Exchange 2000 server needs to read configuration data, such as routing information, it can connect to any domain controller within the local domain to get this information. This is possible because the configuration naming context is replicated to every domain controller in the forest. An Exchange 2000 server will want to use the same domain controller for these requests rather than round-robin.

Through registry values, you can hard-code the names of the directory servers that Exchange 2000 uses to obtain data. This facility is seen as dictating a preference to the Exchange 2000 server. If a given name does not exist or if the server is down, a fail through to standard DNS lookups will take place.

You can use the following registry parameters to dictate the preference for retrieving configuration information:

1.

HKEY_LOCAL_MACHINE \ System \ CurrentControlSet \ Services \ MSExchangeDSAccess \ Instance0

Value name: ConfigDCHostName
Value type: *REG_SZ*
Value data: *DirectoryServer.domain*

(Example: \\BLACK.cat.microsoft.com)

Value name: ConfigDCPortNumber
Value type: *REG_DWORD*
Value data: 0x389

You can use the following Registry parameters to dictate the preference for performing Address Book lookups:

HKEY_LOCAL_MACHINE \ System \ CurrentControlSet \ Services \ MSExchangeDSAccess \ Profiles \ Default

Value name: UserGC1
Value type: *REG_SZ*
Value data: *DirectoryServer.domain*

(Example: \\GINGER.cat.microsoft.com)

Value name: PortNumber
Value type: *REG_DWORD*
Value data: 0x3268 (the standard port number for a GC)

Value name: IsGC
Value type: *REG_DWORD*
Value data: 0x1 (Always set to 1 if the server specified is a GC)

Message Routing and Group Expansion

All message routing information, including routing groups and bridgeheads, is held within the configuration naming context of Active Directory. To determine routing, the Exchange 2000 server contacts a local domain controller and retrieves this information.

If a message is sent to a Universal group, the SMTP virtual server, which is configured to perform the expansion, uses LDAP to contact the Global Catalog and populate the message header with the group membership. If the message is for a Domain Local or Global group, the expansion server uses a local Global Catalog if the membership can be retrieved (for example, if the group is defined within the same domain as the expansion server); otherwise, direct LDAP calls are made to a domain controller in the home domain for the group, and the membership is retrieved remotely. Of course, the latter method requires direct Internet Protocol (IP) connectivity to the remote domain controller.

Address Lists and Offline Address Books

Address Book Views

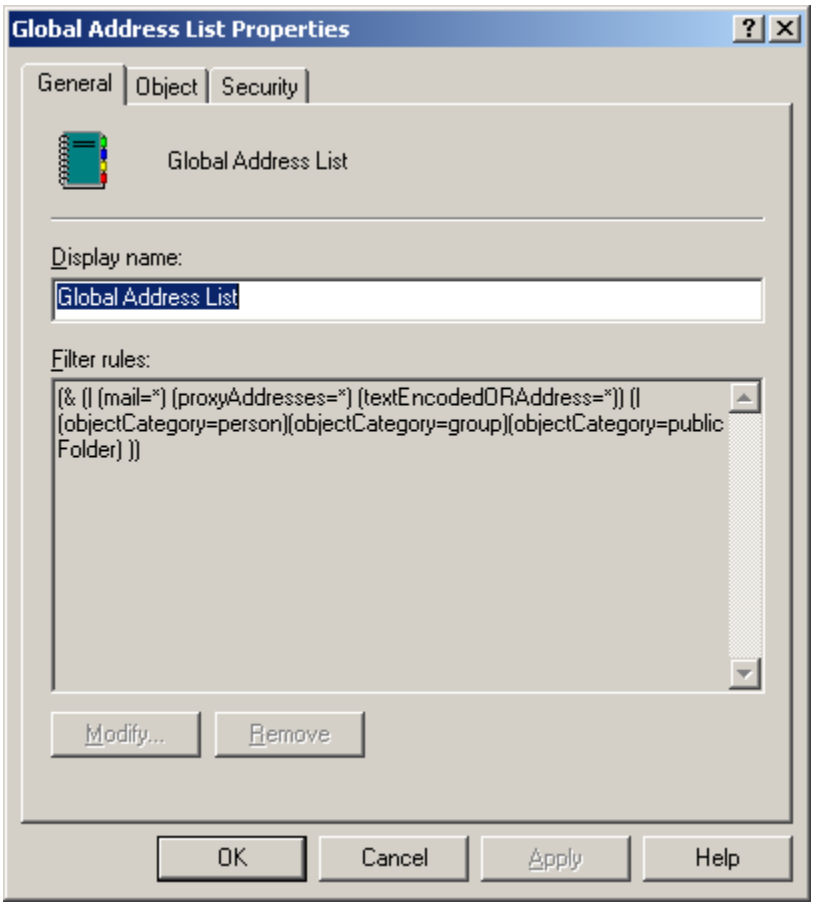
Microsoft Exchange Server version 5.0 introduced the concept of Address Book Views. This facility enables the administrator to expose different views, which are based on field groupings, to the users. With Exchange Server 5.*x*, Outlook users see each site, its containers, and the aggregated Global Address List when they look in the server-based Address Lists. To create a virtual container of all users in the Sales Team, the administrator creates a view grouped by the field **Department**. Although this facility works well, it has various limitations. The biggest drawback is that if you create a view based on a field, a virtual container is created for every unique instance of data within that field. In the previous scenario, you would automatically get other team containers created. This problem exists because the rules that determine how the view is created are not flexible enough for many large organizations.

Address Lists

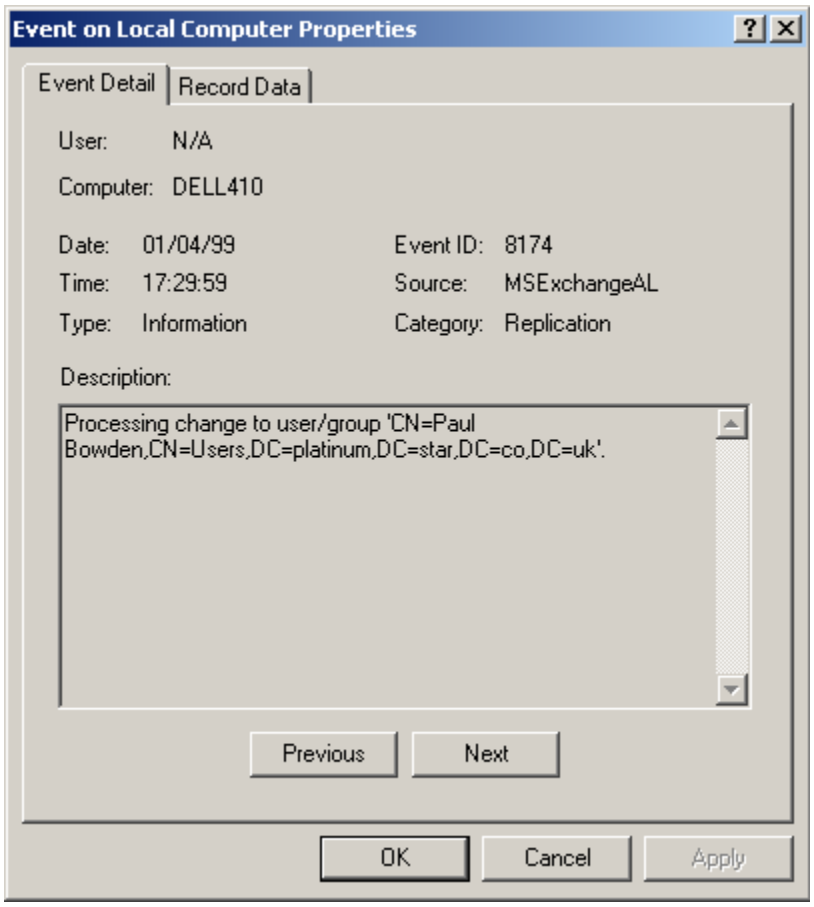
In Exchange 2000, Address Book Views have been superceded by Address Lists. From the Exchange System Manager snap-in, you can create a container that has a build rule associated with it. The rules use the LDAP Search Filter syntax as defined in RFC2254 (see Appendix A) and are extremely flexible. For example, if you want to create an Address List of all permanent employees (not contractors) in the Marketing department in Toronto, you can create a single container called **Marketing** with a rule of:

(&(mail=*)(department=Marketing)(l=Toronto)(!(Extension-Attribute-3=Contractor)))

Unlike Exchange Server 5.5, in Exchange 2000 the Global Address List is not aggregated within Active Directory because it is also built using a rule.



The Exchange System Manager user interface allows you to search with this rule when you create it to ensure that the rule meets your requirements. To ensure that the Address List is up to date, the Address List Service, part of the Exchange System Attendant will poll the directory on a scheduled basis and will populate the Address Lists as necessary. As the user objects are updated, you will notice information events from the MSExchangeAL service with id 8174 in the Application Log, if diagnostics logging is turned up.



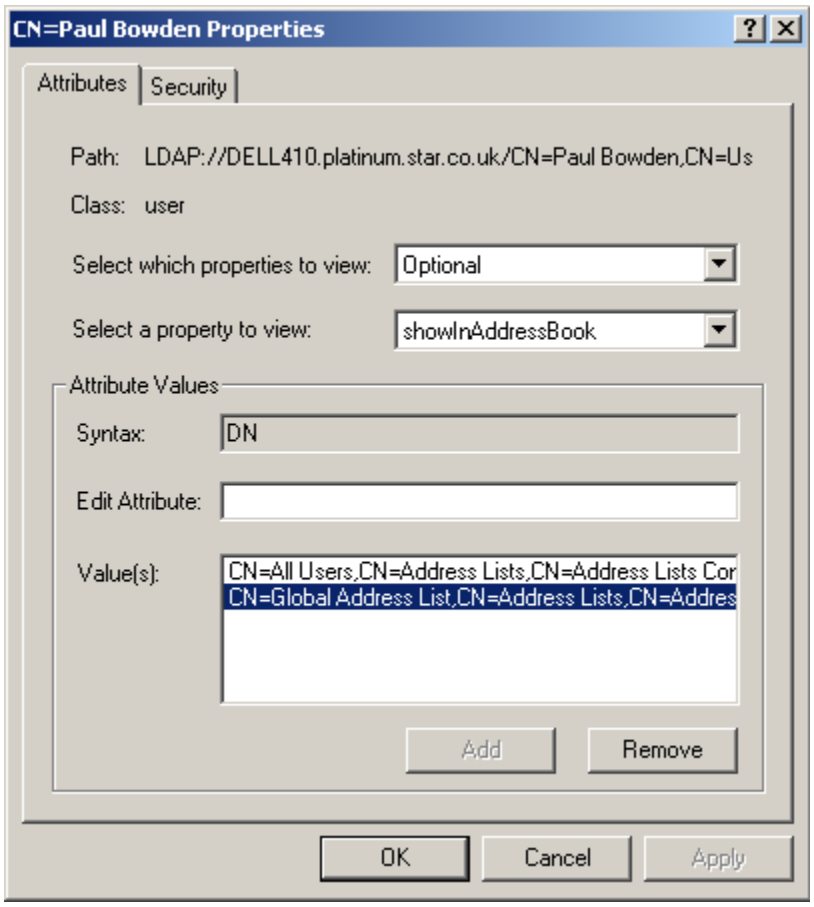
The default Address Lists automatically configured in Exchange 2000 are listed in the following table.

Address List name	***RFC2245 syntax***
Default Global Address List	(&(|(mail=*)(proxyAddresses=*)(textEncodedORAddress=*)) (|(objectCategory=person)(objectCategory=group)(objectCategory=publicFolder)))
All Users	(&(|(mail=*)(proxyAddresses=*)(textEncodedORAddress=*)) (|(&(objectCategory=person)(objectClass=user))))
All Groups	(&(|(mail=*)(proxyAddresses=*)(textEncodedORAddress=*)) (|(objectCategory=group)))
All Contacts	(&(|(mail=*)(proxyAddresses=*)(textEncodedORAddress=*)) (|(&(objectCategory=person)(objectClass=contact))))
Public Folders	(&(|(mail=*)(proxyAddresses=*)(textEncodedORAddress=*)) (|(objectCategory=publicFolder)))
All Conferencing Resources	(msExchResourceGUID=*)

The Address List service populates the defined Address Lists by entering the name and location of the list to the **showInAddressBook** attribute on the user objects in the directory. This data can be seen through utilities, such as Active Directory Services Interface (ADSI) editor or ADSIEDIT, from the Windows 2000 Resource Kit.

1.

To enable ADSIEDIT, copy the ADSIEDIT.* files from the installation CD to the local Winnt\system32 directory and type **Regsvr32 ADSIEDIT.DLL.**



Address List Compatibility with Exchange Server 5.*x*

Upon upgrade of Exchange Server 5.5, all existing Address Book Views within the Exchange directory are migrated to Address Lists in Active Directory and are updated through the Active Directory Connector (ADC). Be aware that Address List updates and new Address Lists are not replicated back to the Exchange 5.x directory.

Address List Service

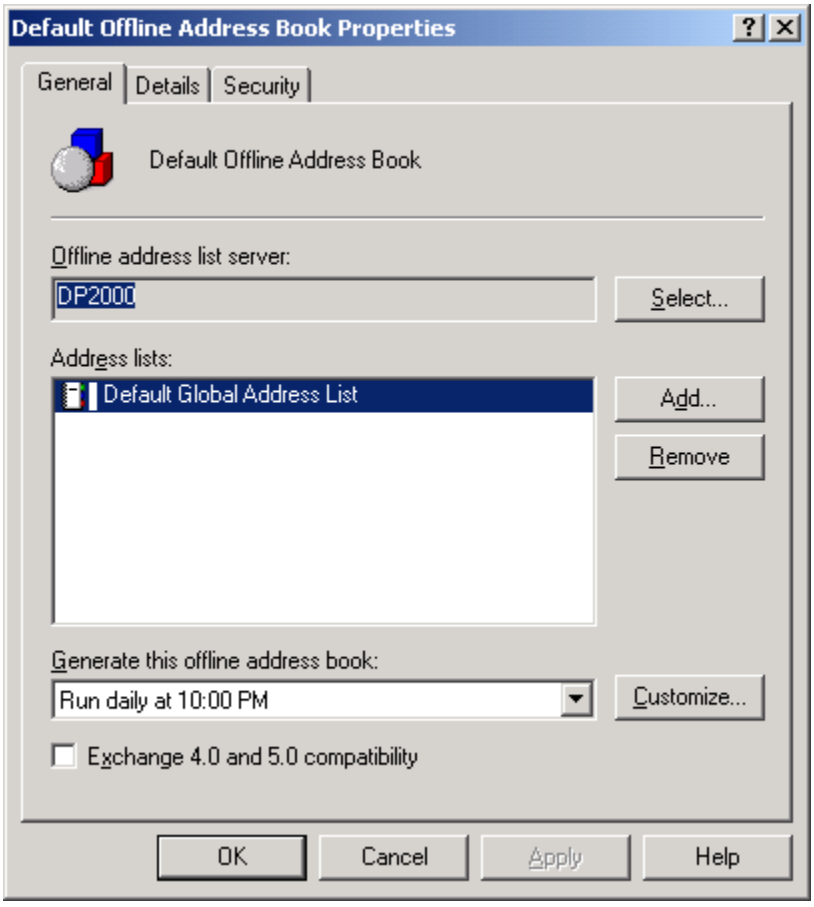
Each Exchange 2000 server can update the Address Lists by making calls to WLDAP32.DLL. Only one Address List service is active within each Active Directory domain; the others remain idle. The service is fully integrated with the Exchange System Attendant (Mad.exe) and as per the set schedule or **Update Now** option, the service will contact a local domain controller and proceed to update the Address Lists based on the rules set.

Offline Address Books

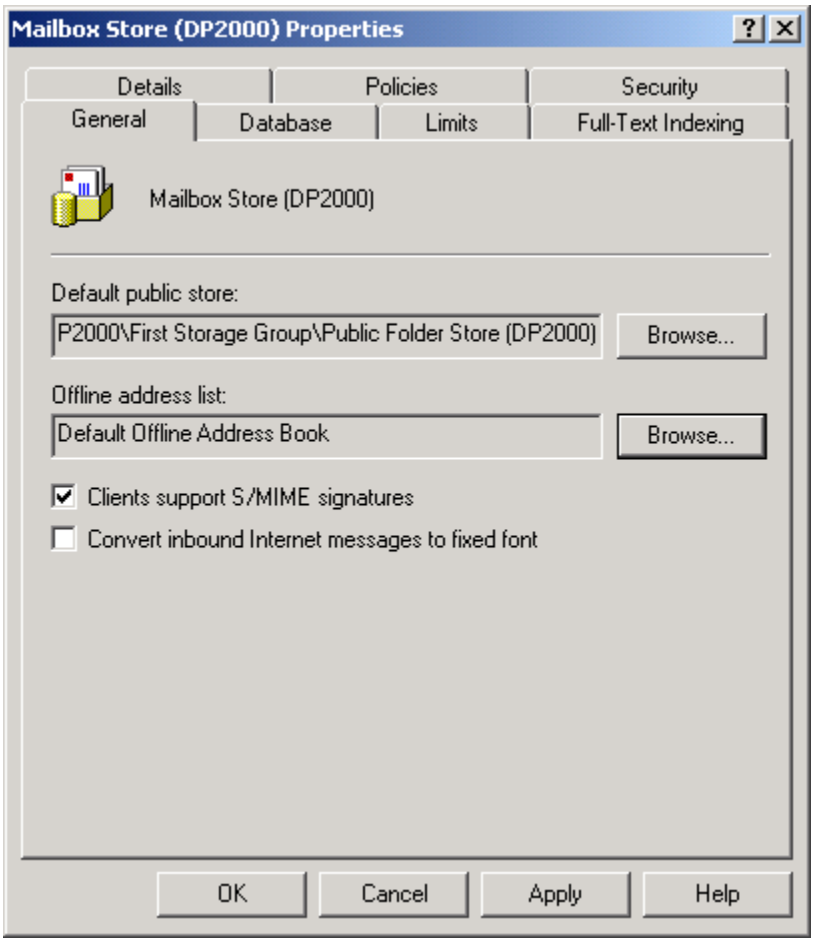
To support mobile users, the Offline Address Book allows users to copy the contents of a server-based Address Book into a set of .oab files on the local user's hard disk. In many companies users need to send e-mail to other users in the company. Subsequently, the Global Address List is normally selected as the source point for the Offline Address List generation.

1.

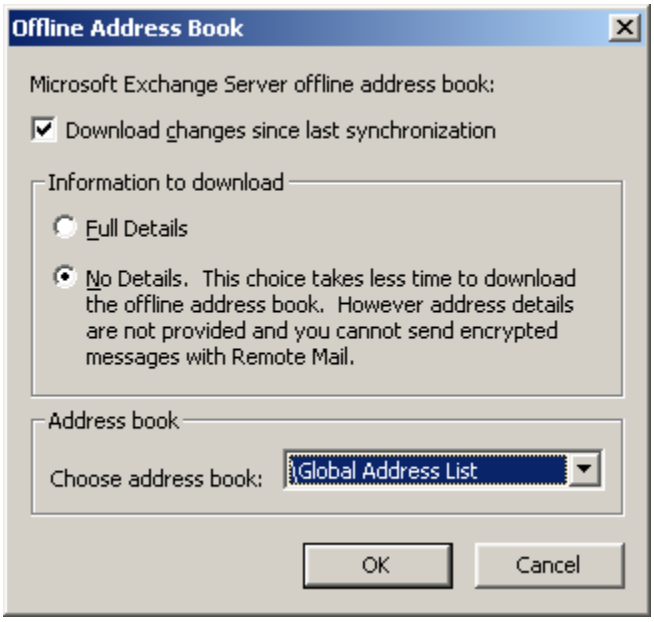
You can configure Offline Address Lists through the Exchange System Manager snap-in. You can select any existing Address List as an Offline Address List. By default, a Default Offline Address Book exists in the Exchange organization.



This Offline Address Book can be associated with private information stores on the Exchange 2000 server, so that Mailboxes located on that server can download the files and consequently, work offline.



When using Outlook 98 and Outlook 97 version 8.03 and later, clients are presented with a list of Offline Address Books to download. Outlook 2000 users can pre-select the Offline Address Book. On the Tools menu, click Settings, and then select **Offline Folder Settings**.


Offline Address Book
Microsoft Exchange Server offline address book:
Download changes since last synchronization
Information to download
Full Details
No Details. This choice takes less time to download the offline address book. However address details are not provided and you cannot send encrypted messages with Remote Mail.
Address book
Choose address book:
Global Address List
OK
Cancel

1.

Appendix A: RFC2254 – LDAP Search Filters

Network Working Group
Request for Comments: 2254
Category: Standards Track

T. Howes
Netscape Communications Corp.
December 1997

The String Representation of LDAP Search Filters

1. Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1997). All Rights Reserved.

IESG Note

This document describes a directory access protocol that provides both read and update access. Update access requires secure authentication, but this document does not mandate implementation of any satisfactory authentication mechanisms.

In accordance with RFC 2026, section 4.4.1, this specification is being approved by IESG as a Proposed Standard despite this limitation, for the following reasons:

a. to encourage implementation and interoperability testing of these protocols (with or without update access) before they are deployed, and

b. to encourage deployment and use of these protocols in read-only applications. (e.g. applications where LDAPv3 is used as a query language for directories which are updated by some secure mechanism other than LDAP), and

c. to avoid delaying the advancement and deployment of other Internet standards-track protocols which require the ability to query, but not update, LDAPv3 directory servers.

Readers are hereby warned that until mandatory authentication mechanisms are standardized, clients and servers written according to this specification, which make use of update functionality, are UNLIKELY TO INTEROPERATE, or MAY INTEROPERATE ONLY IF AUTHENTICATION IS REDUCED TO AN UNACCEPTABLY WEAK LEVEL.

Implementors are hereby discouraged from deploying LDAPv3 clients or servers which implement the update functionality until a Proposed Standard for mandatory authentication in LDAPv3 has been approved and published as an RFC.

2. Abstract

The Lightweight Directory Access Protocol (LDAP) [1] defines a network representation of a search filter transmitted to an LDAP server. Some applications may find it useful to have a common way of representing these search filters in a human-readable form. This document defines a human-readable string format for representing LDAP search filters.

This document replaces RFC 1960, extending the string LDAP filter definition to include support for LDAP version 3 extended match filters, and including support for representing the full range of possible LDAP search filters.

3. LDAP Search Filter Definition

An LDAPv3 search filter is defined in Section 4.5.1 of [1] as follows:

```
Filter ::= CHOICE {
        and                [0] SET OF Filter,
        or                 [1] SET OF Filter,
        not                [2] Filter,
        equalityMatch      [3] AttributeValueAssertion,
        substrings         [4] SubstringFilter,
        greaterOrEqual     [5] AttributeValueAssertion,
        lessOrEqual        [6] AttributeValueAssertion,
        present            [7] AttributeDescription,
        approxMatch        [8] AttributeValueAssertion,
        extensibleMatch    [9] MatchingRuleAssertion
}

SubstringFilter ::= SEQUENCE {
        type    AttributeDescription,
        SEQUENCE OF CHOICE {
                initial        [0] LDAPString,
                any            [1] LDAPString,
                final          [2] LDAPString
        }
}

AttributeValueAssertion ::= SEQUENCE {
        attributeDesc   AttributeDescription,
        attributeValue  AttributeValue
}

MatchingRuleAssertion ::= SEQUENCE {
        matchingRule    [1] MatchingRuleID OPTIONAL,
        type            [2] AttributeDescription OPTIONAL,
        matchValue      [3] AssertionValue,
        dnAttributes    [4] BOOLEAN DEFAULT FALSE
}

AttributeDescription ::= LDAPString

AttributeValue ::= OCTET STRING
```

```
MatchingRuleID ::= LDAPString

AssertionValue ::= OCTET STRING

LDAPString ::= OCTET STRING
```


where the LDAPString above is limited to the UTF-8 encoding of the ISO 10646 character set [4]. The AttributeDescription is a string representation of the attribute description and is defined in [1]. The AttributeValue and AssertionValue OCTET STRING have the form defined in [2]. The Filter is encoded for transmission over a network using the Basic Encoding Rules defined in [3], with simplifications described in [1].

4. String Search Filter Definition

The string representation of an LDAP search filter is defined by the following grammar, following the ABNF notation defined in [5]. The filter format uses a prefix notation.

```
filter     = "(" filtercomp ")"
filtercomp = and / or / not / item
and        = "&" filterlist
or         = "|" filterlist
not        = "!" filter
filterlist = 1*filter
item       = simple / present / substring / extensible
simple     = attr filtertype value
filtertype = equal / approx / greater / less
equal      = "="
approx     = "~="
greater    = ">="
less       = "<="
extensible = attr [":dn"] [":" matchingrule] ":=" value
             / [":dn"] ":" matchingrule ":=" value
present    = attr "=*"
substring  = attr "=" [initial] any [final]
initial    = value
any        = "*" *(value "*")
final      = value
attr       = AttributeDescription from Section 4.1.5 of [1]
matchingrule = MatchingRuleId from Section 4.1.9 of [1]
value      = AttributeValue from Section 4.1.6 of [1]
```

The attr, matchingrule, and value constructs are as described in the

corresponding section of [1] given above.

If a value should contain any of the following characters

Character	ASCII value
*	0x2a
(	0x28
)	0x29
\	0x5c
NUL	0x00

the character must be encoded as the backslash '\' character (ASCII 0x5c) followed by the two hexadecimal digits representing the ASCII value of the encoded character. The case of the two hexadecimal digits is not significant.

This simple escaping mechanism eliminates filter-parsing ambiguities and allows any filter that can be represented in LDAP to be represented as a NUL-terminated string. Other characters besides the ones listed above may be escaped using this mechanism, for example, non-printing characters.

For example, the filter checking whether the "cn" attribute contained a value with the character "*" anywhere in it would be represented as "(cn=*\2a*)".

Note that although both the substring and present productions in the grammar above can produce the "attr=*" construct, this construct is used only to denote a presence filter.

5. Examples

This section gives a few examples of search filters written using this notation.

1.

```
        (cn=Babs Jensen)
        (!(cn=Tim Howes))
        (&(objectClass=Person)(|(sn=Jensen)(cn=Babs J*)))
        (o=univ*of*mich*)
```

The following examples illustrate the use of extensible matching.

```
        (cn:1.2.3.4.5:=Fred Flintstone)
        (sn:dn:2.4.6.8.10:=Barney Rubble)
        (o:dn:=Ace Industry)
        (:dn:2.4.6.8.10:=Dino)
```


The second example illustrates the use of the ":dn" notation to indicate that matching rule "2.4.6.8.10" should be used when making comparisons, and that the attributes of an entry's distinguished name should be considered part of the entry when evaluating the match.

The third example denotes an equality match, except that DN components should be considered part of the entry when doing the match.

The fourth example is a filter that should be applied to any attribute supporting the matching rule given (since the attr has been left off). Attributes supporting the matching rule contained in the DN should also be considered.

The following examples illustrate the use of the escaping mechanism.

```
        (o=Parens R Us \28for all your parenthetical needs\29)
        (cn=*\2A*)
        (filename=C:\5cMyFile)
        (bin=\00\00\00\04)
        (sn=Lu\c4\8di\c4\87)
```

The first example shows the use of the escaping mechanism to represent parenthesis characters. The second shows how to represent a "*" in a value, preventing it from being interpreted as a substring indicator. The third illustrates the escaping of the backslash character.

The fourth example shows a filter searching for the four-byte value 0x00000004, illustrating the use of the escaping mechanism to represent arbitrary data, including NUL characters.

The final example illustrates the use of the escaping mechanism to represent various non-ASCII UTF-8 characters.

6. Security Considerations

This memo describes a string representation of LDAP search filters. While the representation itself has no known security implications, LDAP search filters do. They are interpreted by LDAP servers to select entries from which data is retrieved. LDAP servers should take care to protect the data they maintain from unauthorized access.

7. References

[1] Wahl, M., Howes, T., and S. Kille, "Lightweight Directory Access Protocol (v3)", RFC 2251, December 1997.

[2] Wahl, M., Coulbeck, A., Howes, T., and S. Kille, "Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions", RFC 2252, December 1997.

[3] Specification of ASN.1 encoding rules: Basic, Canonical, and Distinguished Encoding Rules, ITU-T Recommendation X.690, 1994.

[4] Yergeau, F., "UTF-8, a transformation format of Unicode and ISO 10646", RFC 2044, October 1996.

[5] Crocker, D., "Standard for the Format of ARPA Internet Text Messages", STD 11, RFC 822, August 1982.

8. Author's Address

Tim Howes
Netscape Communications Corp.
501 E. Middlefield Road
Mountain View, CA 94043

1.

USA

Phone: +1 415 937-3419
EMail: howes@netscape.com

9. Full Copyright Statement

Copyright (C) The Internet Society (1997). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Appendix B: LDAP Attribute Names in Exchange 5.5 and Exchange 2000

There have been some minor changes to the Lightweight Directory Access Protocol (LDAP) attribute names exposed by Exchange between the two versions. These are listed below along with their associated Messaging Application Programming Interface (MAPI) Identifiers. Any LDAP/ Active Directory Services Interfaces (ADSI) programs that currently use the old syntax for an Exchange Server 5.5 LDAP name must be modified for the new Active Directory syntax.

MAPI ID	Exchange 5.5 LDAP	Active Directory LDAP	Description
0x3A06	givenName	givenName	First name
0x3A11	sn	sn	Last name
0x3A0A	initials	initials	Initials
0x3001	displayName	displayName	Display name
0x3A00	mailNickname	mailNickname	Alias
0x3A29	postalAddress	streetAddress	Address
0x3A27	l	l	City
0x3A28	st	st	State
0x3A2A	postalCode	postalCode	Zip code
0x3A26	co	co	Country
0x3A17	title	title	Title
0x3A16	company	company	Company
0x3A18	department	department	Department
0x3A19	physicalDeliveryOfficeNam e	physicalDeliveryOfficeName	Office
0x3A30	**secretary**	**secretary**	**Assistant**
0x3A08	telephoneNumber	telephoneNumber	Phone
0x8005	manager	manager	Manager
0x800E	reports	directReports	Direct reports
0x3A08	telephoneNumber	telephoneNumber	Business number
0x3A1B	**telephone-office2**	**otherTelephone**	**Business 2 number**
0x3A23	facsimileTelephoneNumber	facsimileTelephoneNumber	Fax number
0x3A2E	telephone-assistant	telephoneAssistant	Assistant number
0x3A09	homePhone	homePhone	Home Number
0x3A2F	**telephone-home2**	**otherHomePhone**	**Home 2 number**
0x3A1C	mobile	mobile	Mobile number
0x3A21	pager	pager	Pager number
0x3004	info	info	Notes
0x8008	memberOf	memberOf	DL membership
0x800F	proxy-addresses	proxyAddresses	E-mail addresses

1.

Note In the Exchange 2000 Beta 3 release, the attributes above marked in blue have no user interface element for setting the data value.